SESSION

INTELLIGENT AGENTS + AUTONOMOUS AGENTS

Chair(s)

TBA

Frequency-Based Patrolling with Heterogeneous Agents and Limited Communication

Tao Mao[1] **and Laura E. Ray**[1]
[1]Thayer School of Engineering, Dartmouth College, Hanover, NH 03755, United States

Abstract - *This paper investigates multi-agent frequency-based patrolling of intersecting, circle graphs under conditions where graph nodes have non-uniform visitation requirements and agents have limited ability to communicate. The task is modeled as a partially observable Markov decision process, and a reinforcement learning solution is developed. Each agent generates its own policy from Markov chains, and policies are exchanged only when agents occupy the same or adjacent nodes. This constraint on policy exchange models sparse communication conditions over large, unstructured environments. Empirical results provide perspectives on convergence properties, agent cooperation, and generalization of learned patrolling policies to new instances of the task. The emergent behavior indicates learned coordination strategies between heterogeneous agents for patrolling large, unstructured regions as well as the ability to generalize to dynamic variation in node visitation requirements.*

Keywords: Frequency-based patrolling, multi-agent systems, decentralized reinforcement learning.

1 Introduction

A persistent patrolling task can be described as the activity of repeatedly visiting specified points in space in order to monitor the surrounding environment or perform a specified function at each point visited. The patrolled area can be described as a graph, in which each vertex or node represents a point to be visited and edges represent possible paths. The problem can be cast according to characteristics of real tasks that are modeled; for example, the number of agents, the number of nodes, and node connectivity may be specified according to patrolling needs and environment constraints. In prior work, it has generally been assumed that each node should be visited with the same frequency, and the resulting multi-agent planning problem is thus characterized as a combinatorial optimization problem with one of several possible criteria defined to maintain uniform visitation; however, it is not always the case that uniform visitation is desirable. Nodes can have different visitation requirements owing to their prominence, patrolled site structure, or activity level within the region [1].

The patrolling problem can be modeled as a coverage problem on closed or open polygons. In prior work, every point on the polygon is generally treated as having equal visitation requirements, and the objective is to either minimize the frequency variance or maximize the average/minimal frequency [2, 3]. The intuitive solution, based on finding a closed polygon for all agents to patrol in the same direction, guarantees uniform visitation and avoids conflicts between agents [2~4]. Elmaliach et al. further develop solutions to this uniform patrolling problem for open polygons [2, 3]. Their work allows agents to overlap trajectories in space but not in time and gives an analytical solution of this model considering real-world uncertainties.

Graph-based patrolling problems, which are abstracted from polygon continuous space patrolling, have also been proposed. In graph-based patrolling with uniform patrol rates for each node, the objective is to either maximize visitation frequency for every point [5] or minimize idle time for every point [6~8]. In models used in these papers, graph edges can have different lengths. For a single agent, the patrolling problem becomes a Travelling Salesman Problem [6]. In multi-agent cases, Hamilton circuit finding is given first priority [5, 6]; if Hamilton circuits do not exist, the algorithms in [5] find the longest path and include outliers, while partition-based strategies are applied in [6]. Reinforcement learning is applied in [7], which seeks an optimal graph partition for each member of the team to patrol. The graph model can be abstracted further in that patrolling efforts are focused on nodes, while edge lengths are ignored and synchronic node visits take place [9]. This abstraction is appropriate for problems involving patrolling of a large region, where the time spent at each node is substantial in comparison to travel time between nodes.

In this paper, we consider the problem of multi-agent patrolling in which the routes patrolled are modeled as undirected circle graphs that intersect occasionally, where one or more agents may patrol each route, and where communication between agents is sparse. As in [9], we focus on patrolled nodes, ignoring edge lengths and assuming synchronic node visitation. The problem is motivated by the need for coordination strategies between heterogeneous agents in patrolling large, unstructured regions. For example, in littoral patrolling, unmanned ground vehicles are constrained to operate on coastal roads, while unmanned

robotic boats may operate on the surface of the water, and unmanned submersibles operate at depths but can also surface. We assume arbitrary visitation frequency requirements for graph nodes.

An objective of solving the patrolling problem described above is to investigate robustness of patrolling policies to variations in the patrolling task, specifically the node visitation frequency requirements and the number of nodes on each graph. Additionally, we wish to examine dynamics that emerge when patrolling policies specific to a given graph model are applied to a variant of that model. For this reason, the problem is cast into a reinforcement learning (RL) framework [10]. The RL algorithm for the patrolling task requires little prior knowledge, and the learned policies are adaptable to the patrolling environment, static or dynamic. However, like other combinatorial optimization methods, RL described as in [11] suffers from "the curse of dimensionality" as the number of nodes and/or agents increases. Thus, we use partially-observable states in order to manage the state space size. The reduction in state space size still permits convergence. Additionally, communication between agents is limited.

The paper is structured as follows: first, we formally define the multi-agent frequency-based patrolling problem targeted in this paper. Next, we review RL and partially-observable states, as applied to this problem. A policy generation method is developed; these policies are exchanged through sparse inter-agent communication to improve the team performance. Finally, we provide empirical results of convergence and communication rates for RL applied to multi-agent frequency-based patrolling. We also evaluate robustness and dynamics of patrolling scenarios in which learned policies generated under a static set of patrolling requirements are applied to a scenario with novel requirements due to variation in node visitation frequency requirements, or number of nodes.

2 Problem definition

Definition 1 *Graph.* [12] A graph is an order pair $G = (V, E)$ consisting of a set of n vertices (or nodes) $V=\{1, 2, 3\ldots n\}$ and a set of $|E|$ edges: $E \subseteq V\times V$. An edge e is a pair $e = (i, j) \in E$. A graph G is undirected if $\forall\ e = (i, j) \in E \Leftrightarrow e' = (i, j) \in E$. A graph G is directed if $\exists\ e = (i, j) \in E$ such that $e' = (i, j) \notin E$

In the patrolling problem, a vertex represents a patrolled node and an edge a possible path choice for the agent.

Definition 2 *General multi-agent patrolling problem.* A group of r agents patrols an area represented by an undirected graph $G = (V, E)$.

According to definition 2, there are n nodes to be patrolled and $|E|$ possible paths for r agents to move. We define the agent to be in position X_i if it is currently visited node i. A *step* of a patrolling agent in patrolling is a move from one node to an adjacent one.

Definition 3 *Circle.* [13] An undirected graph $G = (V, E)$ is considered a circle $C = (V_c,E_c)$ if it has only the set of edges $E= \{(i_1,i_2), (i_2,i_3)\ldots(i_{n-1},i_n), (i_n,i_1)\}$. For simplicity, but without loss of generality, we denote the set of vertex in the circle as V_c, and the set of edges as $E_c= \{(1,2), (2,3)\ldots(n\text{-}1,n), (n,1)\}$.

Definition 4 *Multi-agent frequency-based patrolling.* A group of r agents patrols an area represented by a graph that consists of one or more circles, where each node has a visitation frequency requirement, or patrol rate, F_i. One or more agents can patrol each circle.

The goal of the patrolling task is for the team of agents to minimize the differences between the actual visitation frequency f_i and required frequency of visitation F_i for each node. The patrol route may consist of a single circle graph or multiple, intersecting circle graphs. We abstract dynamics of the vehicles by assuming that the time spent at each node is large compared to the travel time between nodes by each agent. Agents are assumed to communicate only when they occupy either the same node or adjacent nodes so as to model realistic communication limitations in large, unstructured environments.

3 Reinforcement learning for multi-agent Frequency based patrolling

The RL problem, which is Markov Decision Process, can be described as a tuple $\langle R, S, A, p, \rho \rangle$, where R is the set of learning agents; S is the set of possible states; A is the set of actions that the agents can choose from; $p : S \times A \times S \to [0,1]$ is the state transitional probability; $\rho : S \times A \times S \to \mathbf{R}$ is the reward function [10]. RL, Q-learning in particular, defines a state-action value $Q : S \times A \to \mathbf{R}$, which evaluates the value of a certain action taken under a state. Q-learning is a RL algorithm that aims to find satisfactory approximations for Q values during the learning process and uses these Q values to obtain the best action policies for a particular system. The state-action value Q is updated at the end of each episode, and the learned value function Q_t approximates the optimal value function as the algorithm iterates if conditions on the learning rate are met [14]. We use ε-greedy action-value method to determine next actions during learning. This method balances action policy exploration and exploitation.

In the multi-agent frequency-based patrolling problem described in the previous section, the agents can only estimate the frequencies of visitation of nodes and do not know the true values because they cannot obtain other agents' visitation histories except when communication is available. Also, if we choose a state space comprising *every* patrolled node, the

dimension of the learning space increases exponentially with the number of nodes. Considering that each agent moves only to an adjacent node at each step, we propose a partially-observable state comprised of difference frequencies of four nodes nearest to the current node *i* occupied by an agent *r*:

$$s_g \rightarrow s = \left[\Delta f^{i_{-2}}, \Delta f^{i_{-1}}, \Delta f^{i_1}, \Delta f^{i_2} \right] \quad (1)$$

$\Delta f^{i_{-2}}$, $\Delta f^{i_{-1}}$, Δf^{i_1}, and Δf^{i_2} are the difference between the required visitation frequency and the actual visitation frequency for the four nodes nearest to the currently occupied node i_0. In eq. 1, s_g is a global or environmental state while s is a partially-observable state, which we use to model the Markov decision process and train the RL algorithm, for its state space is considerably smaller than that of a global state. At each time step, agents may move either to the left or right, i.e., there are two actions to choose from.

Each agent is provided with the target frequency requirements for all nodes at $t = 0$. During the learning phase, each agent maintains its own visitation history H_j. However, an agent does not know with certainty the visitation frequencies, histories, or policies of other agents. Communication assumptions described previously enable only sparse transmission of information between agents.

4 Policy generation and exchange via inter-agent communication

Since communication is sparse, the information transmitted when communication occurs is important. The state representing the environment is partially observable; hence, the agents should exchange their visitation histories as well as their patrolling policies in order to reduce uncertainty in estimating the aggregate visitation frequencies in eq. 1 at certain time steps.

The theoretical framework for RL is based on properties of Markov Decision Processes. Thus, it is natural to define a policy as a Markov process, which, in the patrolling scenario, is defined as a Markov chain, or a stochastic process with Markov properties. It is assumed that the process has a finite set of states, in which one state transits to another only depending on its current state. The state used in the context of a Markov chain is the position X_i of an agent as in Def. 2, and not the concept of state used for RL. Given historical data, we model an agent's next state to be conditioned on the past states as first-order Markov chain:

$$\Pr\left(X^{t+1} = X_i \mid X \right) = \Pr\left(X^{t+1} = X_i \mid X^t \right) \quad (2)$$

X is a series of states $\{X^t, X^{t-1}, \ldots, X^1, X^0\}$. From this equation, we define the transition probability as

$$p_{ij} = \Pr\left(X^{t+1} = X_j \mid X^t = X_i \right) \quad (3)$$

A transition matrix P for an agent is comprised of elements of probability p_{ij}. The agent updates its transition matrix P from its visitation history using a window of length w time steps. The initial calculation of P is performed after the first w time steps transpire.

A policy π is a stationary policy if it satisfies $\pi = \pi P$ For computation purpose, a stationary policy π can be calculated as

$$\pi = diag\left[\lim_{k \rightarrow \infty} P^k \right] \quad (4)$$

where the *diag*[*] operator retrieves the diagonal elements of the matrix.

Since each agent maintains their own transition policy π_r, they can exchange policies and visitation histories when they communicate. The estimated visitation frequency of a given node can then be obtained both from the agent's own visitation history, other agent's histories when these are available through a prior communication, and the agent's beliefs of others' motions given policies π_{-r} exchanged during communication. Note that when policies are exchanged, they are not necessarily steady-state policies.

At every step, an agent records its own current position, and estimates other agents' positions according to policies π_{-r} exchanged from others. Using both true and estimated histories, the agent calculates visitation frequencies for all nodes. Since estimates of visitation frequencies are based on sampled data, a time window of w_f steps is used to estimate these frequencies. Each agent estimates its current state s_e based on other agents' histories H_{-r} of visitation frequencies (as estimated from exchanged histories and policies π_{-r}) and its own recorded history H_{-r}:

$$s_e = \left[\Delta f_e^{i_{-2}}, \Delta f_e^{i_{-1}}, \Delta f_e^{i_1}, \Delta f_e^{i_2} \right] \quad (5)$$

$\Delta f_e^{i_{-2}}$, $\Delta f_e^{i_{-1}}$, $\Delta f_e^{i_1}$, and $\Delta f_e^{i_2}$ are the difference between the required visitation frequency and the *estimated* visitation frequency for the four nodes nearest to the currently occupied node i_0. Note that visitation histories H_{-r} of other agents are initialized as null, and transitional matrices P_{-r} are initialized as uniform prior to agent communication.

5 Results

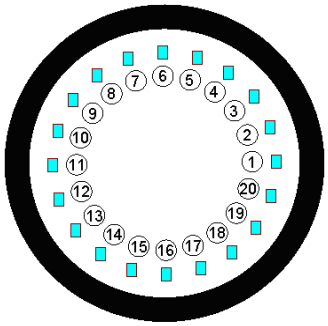
This section presents empirical results based upon two patrolling configurations – a benchmark configuration comprised of a single circle route with uniform patrol frequency requirements (Fig. 1a) and a more general configuration comprised of three intersecting circle graphs,

each having a different number of patrolled nodes and non-uniform patrol frequency requirements (Fig. 1b). Each node has a specified component frequency C_i, and

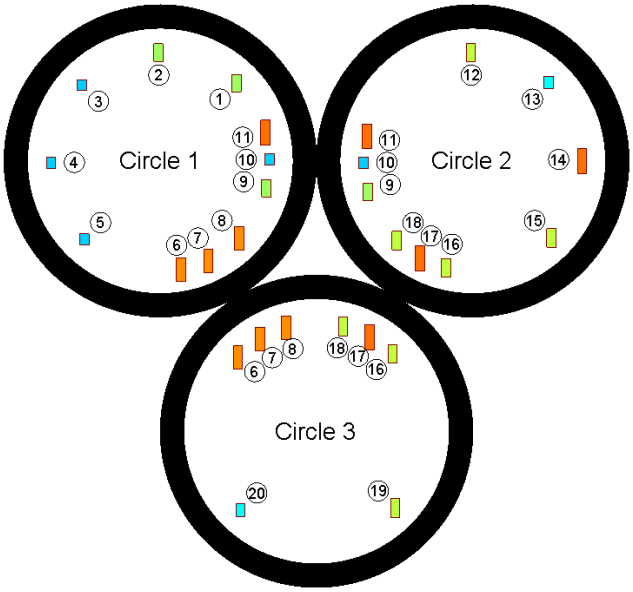
$$\sum_{i=1}^{n} C_i \le 1 \tag{6}$$

Eq. 6 assures that the team has sufficient overall capacity to perform the patrolling task. Our notation for visitation frequency assumes that each agent has a capacity of 100%, and thus the total capacity is 100r%, where r is the number of agents. Then, the visitation frequency requirement for Node i is defined as $F_i = 100rC_i\%$.

Q-learning is used in the learning phase, with an ε-greedy criterion for action selection. ε is initialized as 1 and exponentially decays to 0.01. An agent receives a reward of f_i - F_i when it occupies Node i.



(a) Benchmark configuration



(b) General configuration

Figure 1. Patrolling configurations

Time window parameters w and w_f are 1000 and 100 steps respectively. We assess the efficiency of the algorithm by measuring the difference between the required and actual patrol rate at each node as a function of learning step. We formally define the insufficiency of the patrol rate I_i for Node i as a performance metric:

$$I_i = \begin{cases} F_i - f_i & , F_i - f_i > 0 \\ 0 & , F_i - f_i \le 0 \end{cases} \tag{7}$$

Note that this measure does not penalize the use of excess capacity, i.e., when the actual patrol frequency exceeds the desired patrol frequency, sufficiency requirements are met and thus insufficiency of patrol rate is zero.

5.1 Single-circle multi-agent patrolling

The benchmark configuration of Fig. 1a has n = 20 nodes, r = 2 patrolling agents, component frequency C_i = 4.5% for each node, and uniform patrolling frequency requirements of F_i = 9% at each node; thus, the system is overmanned in the sense that collectively, agents have sufficient capacity to meet visitation frequency requirements. This configuration is designed to test the learning algorithm in the case where there exist multiple, obvious solutions. Here, the range of policies that meet a uniform patrol frequency requirement varies: agents each traversing the route in a single direction, splitting the graph equally between agents, or agents each making random moves with uniform probability would all comprise solutions meeting this requirement. In the absence of additional criteria, all solutions are satisficing, or acceptable solutions. A RL approach generally converges to more stochastic than deterministic solutions when two actions are selected with equal probability during learning exploration.

We perform 10 experimental runs of 10^6 learning steps each. In each run, agents learn a patrolling policy from scratch. Each learning step corresponds to a time step owing to abstraction of travel between nodes. Figure 2 shows mean and 1-sigma variation of $\sum I_i/n$ over 10 experimental runs and 20 nodes. Figure 2 also shows the number of instances of communication between two or more agents out of the last 1000 steps averaged over 10 experimental runs. These measures are shown as a function of learning step. Results show that a patrolling policy that achieves a visitation frequency within 1% of the target frequencies over 10 experimental runs emerges, providing a satisficing policy in about 200,000 steps. Three stages of convergence are evident. In the "adjustment" stage (< 10,000 steps), agents explore a new environment, collect experience for learning, and communicate sparsely. In the "convergence" stage (10,000 ~200,000 steps), learning starts to improve performance, reducing the average insufficiency of patrol rate. Finally, in the "converged" stage (> 200,000 steps), a satisficing policy emerges, and there is little room for the two-agent group to

improve performance. Communication plays an important role in learning convergence as the increase in communication rate starting at ~10,000 steps helps the agents estimate their belief states with less uncertainty and makes Q-learning converge. In addition, the communication frequency has changed from random to more regular, as indicated by the communication pattern shown as the two small figures in Fig. 2. In the converged policy, each agent chooses to move left and right with nearly 50% frequency. Thus, each agent's policy comprises a roughly 50-50 chance of going left or right, which is a reasonable solution for this benchmark problem.

5.2 Multi-circle multi-agent patrolling

In the general configuration, 20 nodes are distributed in three circles for three agents to patrol, as in Fig. 1b. Some nodes are common to two circles and appear twice in the figure. Component frequencies for the 20 nodes are assigned non-uniformly, while obeying the constraint imposed by eq. 6. The component frequencies are 4.33%, 4.33%, 2.67%, 2.67%, 2.67%, 6.00%, 6.00%, 6.00%, 4.33%, 2.67%, 6.33%, 4.67%, 3.00%, 6.33%, 4.67%, 4.67%, 6.33%, 4.67%, 4.67%, 3.00%, and required visitation frequencies are three times component frequencies. The sum of component frequencies is 90%, indicating that the team has sufficient capacity. In Fig. 1b, the length of the bars inside the circles associated with each node indicates the resulting F_i, as does the color, with blue indicating lowest F_i and red indicating highest F_i.

We perform 10 experimental runs of 10^6 steps each, in which agents learn from scratch for each run. Figure 3 shows mean and 1-sigma variation in the mean insufficiency of patrol rate averaged over 20 nodes and 10 experimental runs, as well as the number of instances of communication between two or more agents out of the last 1000 steps averaged over 10 experimental runs, as a function of learning step. A satisficing policy that achieves average insufficiency of patrol rate below 1% is learned in ~10^5 steps. As in the benchmark configuration, there are three stages in convergence of the learning algorithm. Improved convergence rate over the benchmark problem is attributed to the lower number of patrolled nodes per agent.

Figure 4 shows snapshots of the visitation policy convergence for each patrolling agent in the form of a "heat map" designating the frequency of occupancy of each agent on the graph and a colored bar for each node. The color in a bar indicates the visitation requirement as specified in Fig. 1b. The length of color filled in each bar denotes the patrol rate agents provide to a node in order to fullfil the frequency requirement. For overlapped nodes, the sum of each agent's contribution to patrolling a node, denoted by the color of the associated point in the circle, produces the total actual visitation frequency. Prior to convergence, visitation requirements are not satisfied at every node, and agents do not exhibit strong cooperation. Subsequently, the agents begin to increase their communication rate as seen in Fig. 3; cooperation emerges; and the patrolling requirements are met. For instance, Agent 1 has the heaviest workload along its circle (11 nodes) and Agent 3 has a comparably lighter workload (8 nodes). Thus, in the overlapped nodes on these two graphs, Agent 3 takes most of the workload. Policy convergence is achieved with cooperation.

5.3 Dynamic multi-agent patrolling

Due to the number of learning steps required to establish a patrolling policy, it is desirable that a learned policy generalize to variations in node visitation requirements and number of nodes. Here, we evaluate the transient dynamics and steady-state performance when the learned policy is applied to each of two variants of the general configuration. In the first variant, frequency requirements for Node 4 and Node 14 are swapped, with Node 4 requirements increasing from from 8% to 19% and Node 14 decreasing from 19% to 8%; and in the second, a scenario of adding and deleting a patrol node is obtained by setting the frequency requirement for Node 2 from 13% to zero and adding a new node Node 21 between Node 19 and 20 with a frequency requirement 14%.

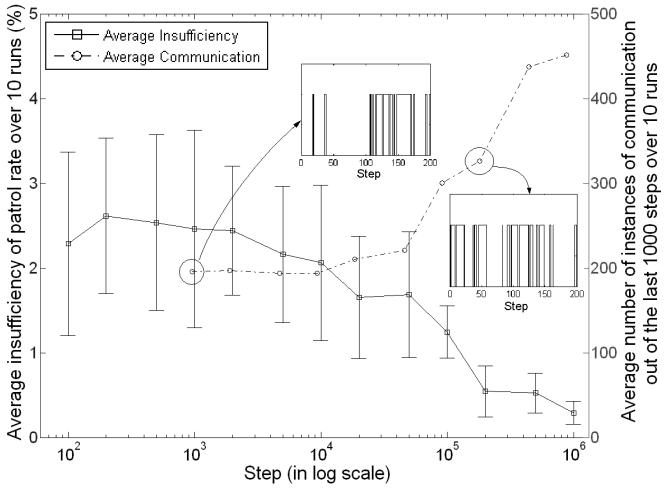


Figure 2. Learning convergence of benchmark configuration

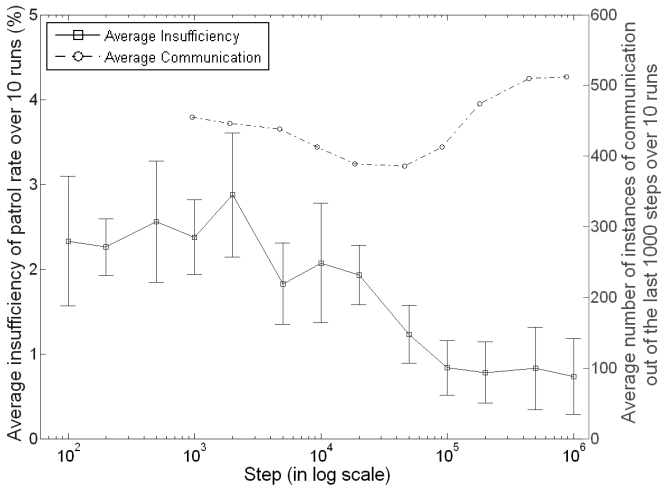


Figure 3. Learning convergence of general configuration

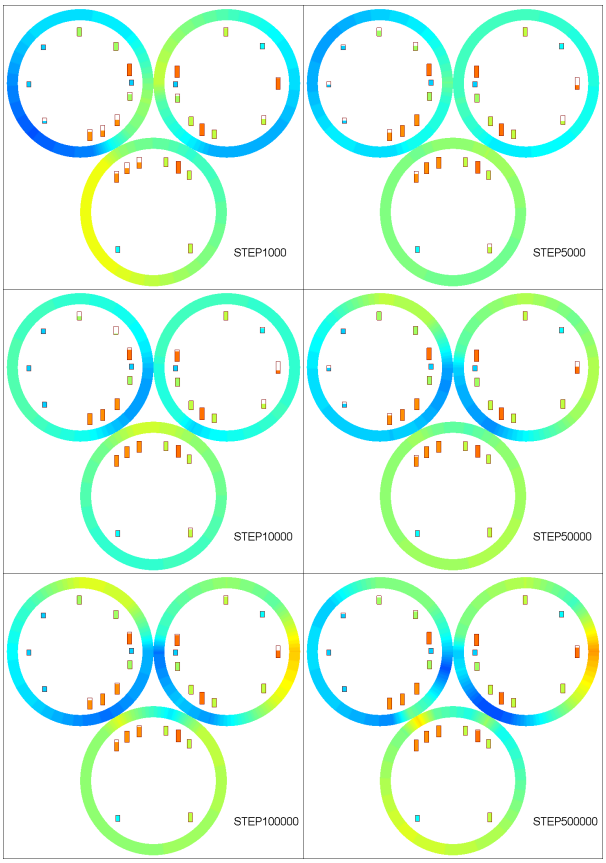


Figure 4. Snapshots of learning in the general configuration

Figure 5 indicates transients in $D_i = f_i - F_i$, the difference between actual and required visitation frequency at each node for each experiment. In the first experiment (Fig. 5a), the agents reduce the frequency difference range from between -11.1% and 11.4% to between -2.1% and 5.3% within 100 steps. After 100 steps, most transients decay to within in the color range of green, yellow or red, indicating frequency requirements of these nodes are met or exceeded. In the second experiment (Fig. 5b), the agents reduce the frequency difference range from between -14% and 13.4% to between -3.1% and 4.7% in 100 steps. After 100 steps, most transients decay to within in the color range of green, yellow or red, which indicates frequency requirements of these nodes are met. These examples indicate that the collective learned policies can be applied to a new instance of the task in a dynamic environment with new nodes.

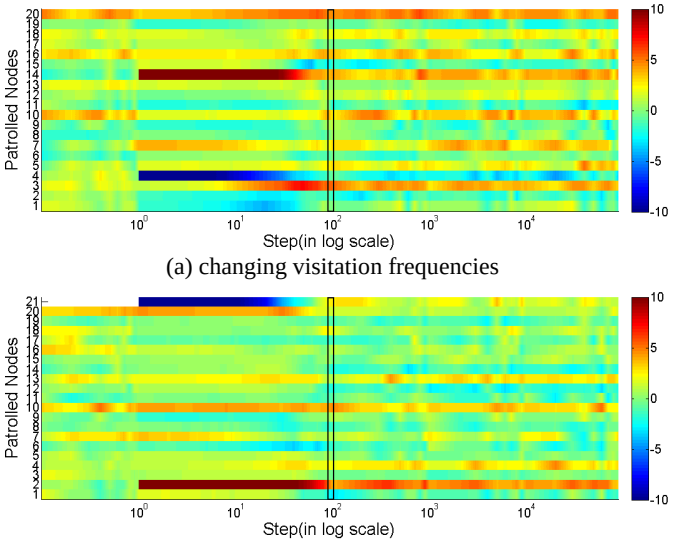
Figures 6 and 7 show dynamic evolution of the heat map for each of these experiments. In Fig. 6, cooperation emerges between Agents 1 and 2; due to workload reduction for Node 14, Agent 2 increases visitation of Node 11 to help Agent 1 meet the increased demand from Node 4. In Fig. 7, cooperation emerges between three agents: Agent 1 assists Agent 3 by assisting Agent 2. Due to zero workload for Node 2, Agent 1 applies effort to visit Nodes 9~11; then, Agent 2 has the capacity to help Agent 3 alleviate the workload on Nodes 16~18; finally, Agent 3 can handle new Node 21. In other words, indirect cooperation between Agent 1 and Agent 3 emerges through Agent 2's involvement. Node 2, which requires no visits, is not avoided due to its physical existence in Circle 1; however, the visitation frequency for this node is reduced to 4%.

Comparison of steady-state heat maps for dynamic variations of the task from Fig. 6 and 7 with the heat map for the standard configuration at convergence of learning (Fig. 4, snapshot at step 500,000) shows that a range of dynamics can be elicited from a single-run learned policy.

6 Conclusions

We apply Q-learning based on partially-observable states to a multi-agent frequency-based patrolling problem where frequency requirements for each node can be non-uniform, and communication is limited. Transition matrices describing the probability of state transitions are derived from Markov chains, and policies are exchanged when two agents occupy the same or adjacent nodes. In both a benchmark configuration with uniform patrol frequency requirements and a multi-graph patrol configuration with non-uniform requirements, multi-agent teams learn patrolling policies embodying cooperation. Empirical results demonstrate that the patrolling policies learned from static configurations can be applied to new instances of the patrolling task, with variations in the number of nodes or changes in patrol frequency requirements.

The relaxation of uniform frequency requirements and modeling of communication as sparse makes the model realistic for patrolling large, unstructured regions using heterogeneous agents. Additionally, the more stochastic policies derived from RL can make it difficult for external agents to predict patrolling agents' behaviors, which is a desirable feature for many applications.



(a) changing visitation frequencies

(b) adding and deleting nodes

Figure 5. Transients in difference between actual and required visitation frequency at each node in dynamic environments

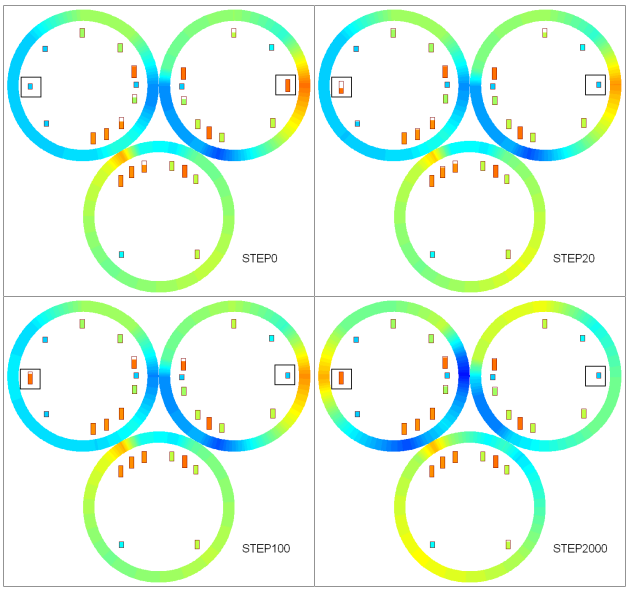


Figure 6. Policy transition in dynamic environment with changing visitation frequencies

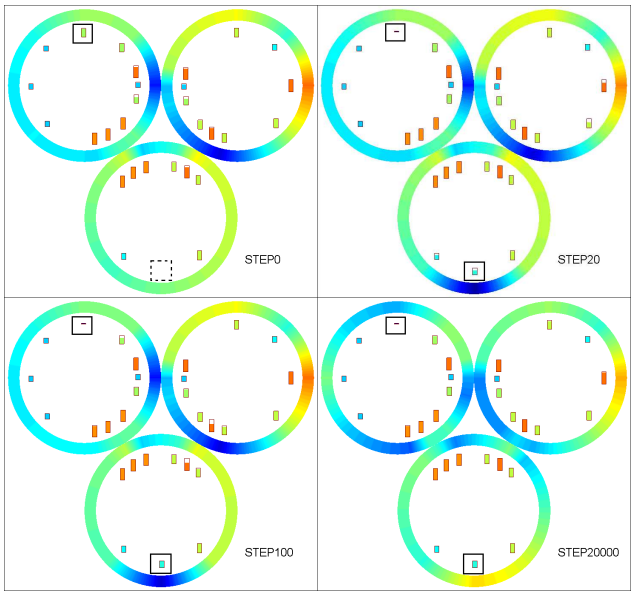


Figure 7. Policy transition in dynamic environment with adding and deleting nodes

7 Acknowledgements

This work is supported by the Office of Naval Research under Multi-University Research Initiative (MURI) Grant No. N00014-08-1-0693.

8 References

[1] Almeida, A., Ramalho, G., Santana, H., and et. al. "Recent advances on multi-agent patrolling"; Advances in Artificial Intelligence-SBIA, 3171, pp. 526-535, 2004.

[2] Elmaliach, Y., Shiloni, A., and Kaminka, G. "A realistic model of frequency-based multi-robot polyline patrolling"; Proceedings of the International Joint Conference on Autonomous Agents and Multi-Agent System, pp. 63-70, 2008.

[3] Elmaliach, Y., Shiloni, A., and Kaminka, G. "Frequency-based multi-robot fence patrolling"; Technical Report, Computer Science Department, Bar Ilan University, Israel, 2008.

[4] Elmaliach, Y., Agmon, N., and Kaminka, G. "Multi-robot area control under frequency constraints"; Proceedings of IEEE International Conference on Robotics and Automation, pp. 385-390, 2007.

[5] Portugal, D. and Rocha, R. "MSP algorithm: multi-robot patrolling based on territory allocation using balanced graph partitioning"; Proceedings of 25th ACM Symposium on Applied Computing, Special Track on Intelligent Robotic Systems, pp. 1271-1276, 2010.

[6] Chevaleyre, Y. "Theoretical analysis of the multi-agent patrolling problem"; Proceedings of IEEE International Conference on Intelligent Agent Technology, pp. 302-308, 2004.

[7] Perron, J., Moulin, B., Berger, J., and et. al. "A hybrid approach based on multi-agent geosimulation and reinforcment learning to solve a UAV partolling problem"; Proceedings of the 2008 Winter Simulation Conference, pp. 1259-1267. 2008.

[8] Machado, A., Almeida, A., Ramaldo, G., and et. al. "Multi-agent movement coordination in patrolling"; Proceedings of 3rd Intl' Conf. on Computers and Games, 2002.

[9] Baglietto, M., Cannata, G., Capezio, F., and Sgorbissa, A. "Multi-robot uniform frequency coverage of significant locations in the environment"; Distributed Autonomous Robotic Systems, Vol. 8, pp. 3-14, 2009

[10] Sutton, R. S., and Barto, A. G. "Reinforcement Learning: An Introduction." Cambridge, MA: MIT Press, 1998.

[11] Bellman R.E. "Dynamic Programming". Princeton, NJ: Princeton University Press. 1957.

[12] Lovasz, L., Pelikan, J., and Vesztergombi, K. "Discrete Mathematics". New York, NY: Spring-Verlag Press, 2003.

[13] Diestel, R. "Graph Theory (translation of Graphen-theorie)". New York, NY: Spring-Verlag Press, 1996.

[14] Watkins, C. J. "Q-learning"; Journal of Machine Learning, Vol. 8, pp. 279-292. 1992.

Extending the General Game Playing Framework to Other Languages

Xinxin Sheng, David Thuente
North Carolina State University

Abstract

General Game Playing (GGP) research aims at building automated intelligent computer agents that accept declarative descriptions of arbitrary games at run time and are able to use such descriptions to play effectively without human intervention. This paper extends the Game Description Language (GDL) to more easily describe complicated games, including social and financial games with diverse goals. The GDL extension allows adding functions, defined in other programming languages such as Java, C++, Lisp, to the general game playing architecture. We used the three-player, simultaneous financial game Farmer, a standard GGP game, as an example to illustrate the need for and the benefits of the GDL extension. We also show the extension of the GDL is critical for adding coalition games to the general game playing area.

1 Introduction

General Game Playing (GGP) research aims at building automated intelligent computer agents that accept declarative descriptions of arbitrary games at run time, and are able to use such descriptions to play effectively without human intervention. GGP as a research topic was started at Stanford University in 2005. The GGP competition has been held annually at the AAAI conference since then. Many successful GGP players have emerged [Clune, 2007; Kuhlmann and Stone, 2007; Finnsson and Bjornsson, 2008; Schiffel and Thielscher, 2007; Sheng and Thuente, 2010a; Mehat and Cazenave, 2011; Möller et al., 2011].

GGP is designed to include a wide variety of games, including one-player games, two-player games, multiple-player games, turn-taking or simultaneous games, zero-sum or non-zero-sum games, etc. The players can have conflicting interests or compete for scarce resources. The GGP design should allow coalition games as well but they have not been addressed as such and are not included. There are about 40 kinds of games in the GGP database with more than 200 variations that are created by slight modifications. Many of these games have been played in the AAAI general game playing competitions. For these competitions no human intervention is allowed, no matter whether the agent plays a new game that the world has never seen before, or an old game that has already been solved. The agents have to automatically adapt not only to play the games but obviously more so, to play well.

The General Game Playing framework is a multi-agent system with one game server agent coordinating several game playing agents. The player agents communicate through the game manager but not with each other. The game manager starts the automated game play by sending out the game information and setting up the playing time. The game manager provides the player agents with the initial state of the game, the set of game rules, and the agent's own role in the game. With such information, the agents use knowledge reasoning to calculate what are the legal actions to take.

The GGP games are written in Game Description Language (GDL) [Love *et al*., 2005], which is written in KIF [Genesereth and Fikes, 1992], a subset of First Order Logic (FOL). GDL can be understood as a specification language for a large class of multi-agent environments, working as formal laws to guide the agents to understand the game rules and their own roles so that they can participate in the games [Schiffel and Thielscher, 2009]. All agents participating in the game accept game rules as provided and take proper actions to win. The game manager validates and broadcasts all actions submitted by different player agents and thereby moves the game forward to the next stage.

A game played in the GGP framework can be understood as a finite state transition system with one initial state and at least one terminal state. The game rules are the transition functions between the states. A game starts from the initial state and transits from one state to the next precisely in response to the actions taken.

GDL grammar and syntax can be used to define functions, constants, variables, operators, game states and game rules. Constants can be relation constants such as *adjacent*, or object constants, such as *King* (in chess). The types of constants are determined by the context. Variables begin with a ? and the values are determined in the logic resolution process. All operators are in prefix notation, including the negation. The state of the game is described with a group of propositions that are true in that state. A game rule is a prefix implication. The head of the rule is a relation constant with a number of arguments. The body of the rule contains zero or more literals, or negative literals. Recursion is allowed, but with limitations to avoid unbounded growth.

The Game Description Language is quite powerful and has defined numerous different games. However, as a language written in the First Order Logic, GDL suffers the limitation of

the FOL as well. This paper explains the limitation of the current framework of the GDL in Section 2. The extension of the GDL is proposed in Section 3. And the application usage of the extension is illustrated in Section 4.

2. Limitations of the GDL

Game playing is limited by time and space resources. Most games are not exhaustively searchable. The agents need to allocate the limited time to logic reasoning, game tree search, the action payoff evaluation, machine learning, and game states transition, etc. Logic reasoning participates in all these operations. Small performance improvements in the reasoning process can be greatly amplified to expedite searches and learning and can therefore generate huge advancements in the agent's performance. We have proposed using hash tables to improve the reasoning efficiency in previous research [Sheng and Thuente, 2010b].

In addition to the logic resolution data structure, we found out how a game is written affects the reasoning performance as well. For example, to determine if there exists a line in Tic-Tac-Toe, the function can be written either as "find out all lines on the board, and check if one is owned by myself", or as "find out all my pieces, and check if any three formed a line". The computational complexity is different in these two expressions. In big games, some expressions can become so computationally demanding that the desired function can not be reasonably defined.

2.1 The Financial Game Farmer

The Farmer game is a three-player simultaneous non-zero-sum financial game especially written for GGP. Traditional computer game research covers mainly two-player turn-taking games, but the Farmer game sets up tasks for three players (*Alice*, *Barney* and *Charlie*) to make as much money as possible in a certain number of iterations (there are 10, 20, 40 iterations versions in GDL of Farmer game in which the game terminates at the designated iteration and rewards are calculated). Each player tries to maximize his own net worth. We found this game very interesting and there are several publications discussing it [Sheng and Thuente, 2011; Sheng and Thuente, 2010a].

The three players in Farmer each start with 25 gold pieces. The players begin with no commodities on hand. The initial market price for wheat is 4 gold pieces, flour 10, cotton 7 and cloth 14 gold pieces. After every step of the game, the market price of each commodity automatically increases by 1 even if nothing is bought or sold. In addition to the automatic increase, the product price of that commodity decreases by 2 in the next iteration for every distinct selling action (maximum decrease of 6 in one step for the three player game) of that commodity, and increases by 2 for every buying action of that commodity.

The Farmer game contains a simplified skeleton of the manufacturer financial activities, such as investing in infrastructure (buy farm or build factory), buying raw material from the market, selling the product to the market, and keeping the inventory down, etc. It also contains the elements of the stock market, such as an inflation indicator (price increases by 1 in every step.), price affected by market activities (price goes down by 2 for every selling action, and up by 2 for every buying action), the player with the most money wins, etc. The Farmer game opens the possibility of using the general game playing techniques to explore and perhaps even solve skeletal real world financial or social problems.

2.2 Burden of the Natural Numbers

In a financial game with items to buy and sell, arithmetic addition and subtraction are used to calculate the gold amount. Natural numbers are not defined as part of the GDL. Instead, the arithmetic operations are defined as part of the game rules. Figure 1 is part of the Farmer game that defines the addition operations of natural numbers. Bold indicates the first definition of a function.

```
1.  (number 0) //define unary relation number
2.  (number 1)
3.  ...
4.  (number 300)
5.
6.  (succ 0 1) //define the sequential relation
7.  (succ 1 2)
8.  ...
9.  (succ 299 300)
10.
11. (<= (sum ?x 0 ?x) //define function addition
12.      (number ?x))
13. (<= (sum ?x ?y ?z) //define recursive function addition
14.      (succ ?x ?m)
15.      (succ ?n ?y)
16.      (sum ?m ?n ?z))
```

Figure 1. Arithmetic addition defined with GDL in the Farmer game.

The Farmer game defines the unary relation number, and the order sequence of the numbers. The sum function in line 11 defines {x+0 = x}, and line 13 is the recursion version of the sum function. In this piece of code, only natural numbers between 0 and 300 are defined. The addition calculation is extremely tedious. To calculate (1+299), for example, the game rules as defined in line 13 need to calculation (2+298) in line 16, then (3+297) *etc.* After performing 300 layers of recursion, the result {(300+0) = 300} can be calculated from the base definition in line 11, and the stack is popped until (1+299) value returns. In this schema, it is not even possible to calculate (2 +300). Even though both numbers are defined in the game, the sum is not defined.

Suppose we can define such a big set of numbers that we are completely confident that it is enough for the game to use. In this case, let the numbers defined be *n*, the time complexity for the number or succ function are O(*n*), so the recursion

addition calculation as defined in line 13 has the time complexity of $O(n^2)$. A simple arithmetic addition operation is expected to be conducted in O(1) time. The current GDL structure is obviously not satisfactory in terms of computational requirements.

2.3 Confusion of the Comparison

Although natural numbers are not defined as a GDL component, they are used in the goal definition loosely in many games. For example, in Tic-Tac-Toe, when a player forms a line, it is rewarded with 100 points with the rule (<= (goal ?player 100) (line ?player)), and is rewarded with 0 points when the opponent forms a line. In both situations, the numbers were used without being defined, how can the agent realize 100 is more desirable than 0?

In logic operations, which term is used does not affect the logic relations as long as the changes are consistent. Anything that is not defined as a keyword in the GDL is substitutable. In fact, in the GGP competitions, to avoid human cheating, all the competition games were scrambled into random, meaningless letter combinations. The agent, operating correctly, can perform the same logic inference on the scrambled version. As long as the logic relations are the same, the scrambled version of the game will be treated the same. For example, the Tic-Tac-Toe state description (cell 1 1 blank), which means a ternary relation cell that relates a row number, a column number, and a content designator, is logically equivalent to the description (position first first empty), or scrambled version (tsmpzz ryy ryy ffgh), as long as the terms are replaced consistently in all other occurrences of the game rules [Sheng and Thuente, 2010b].

Since substitution is not supposed to change the logic relations, the GDL code (<= (goal ?player 100) (line ?player)) can be substituted to (<= (goal ?player sss) (line ?player)) as long as all the definitions and usages of the constant 100 are substituted with sss universally. In many games, 100 (or 0) is used without definition. Although the numerical goal rewards have never be scrambled in the past competitions, being not defined as keywords but used as regular logic symbols, they are subjected to the scramble operation just like any other symbol. Now the agent is facing a logic resolution loophole: is the goal sss more desirable than the goal ttt?

In some situations where the players are given partial credit, such as moving four pieces into the designated place and getting 25 points for each piece, it is not possible to choose one goal over another without knowing the values of the goals.

To define the goals more rigorously, the game need to define unary relation *number* as in Figure 1. Then define a *greater* function for the agent to compare the value of 100 and 0. The agent can order all the goal definitions and determine which one is the most desirable. Only in this way, can the agent calculate the partial credit for completing partial tasks too.

2.4 Challenge of Infinity

The finite set of natural numbers is hard enough for the GDL to handle. When the content of the game element involves an infinite set, the existing structure faces new challenges.

In the Farmer game, the market price of each commodity automatically increases with the inflation factor every iteration as time goes by. Suppose the inflation factor is 10% of the current price. Then the wheat price with no transaction influence is 4 gold in the first step, 4.4 in the second, 4.84 in the third, 5.324 in forth, and so on.

In the GDL, it is still possible to define the real number set by defining the computational elements, such as the carry flag, the borrow flag, the negative sign, etc. But for the real number infinity set, there does not exist a reasonable way to describe such a scenario.

We can go on with the limitations of GDL and how to use it better to define an infinity set by proposing a overflow threshold just like all computer systems do. But the discussion of how to define more powerful scenarios with the GDL misses the point. The GDL structure is for accommodating heterogeneous games in one multi-agent system for the player agents to play. Re-writing of these computer games with GDL is not only cumbersome but also not necessary.

3. Extension of the GDL

GDL describes the basic game elements such as game initialization (keyword *init*), state transition (keyword *next*), legal actions (keyword *legal*), game roles (keyword *role*), game goals (keyword *goal*), game termination (keyword *terminal*), etc. Researchers have written about forty games with more than 200 variations with the GDL.

The GDL has never meant to be a closed system. There have been efforts to extend the GDL to describe new groups of games. New keywords *sees* and *random* were introduced in order to add incomplete information games [Thielscher, 2010]. New keywords *message, receive, send, time* were added to describe market trading scenarios [Thielscher and Zhang, 2009]. Adding new keywords and a new group of games to the GGP family certainly enriches this area. However, the process can be generalized in a manner that makes new game definitions much easier, simpler, and clearer.

We propose adding the keywords *include* and *use* into the GDL. Just like importing packages in the Java language, or including the header files in C++, the *include* keyword in GDL serves as an interface allowing area specific definitions and functions to be shared and quoted without affecting the generality of the GDL. The *include* keyword serves as the information facade that bridges the game logic representations described with GDL and the area specific contents. The keyword *use* is to call external functions with parameters.

The major contribution of the extension is that an existing game does not have to be completely re-written into GDL for the general game player to play. A game does not have to be written in one computer language either. Certain game information (see section 5 for more details) can be defined and

calculated in the user's favorite programming languages and passed back and forth through a command line interface.

3.1 Implementation of the GDL Extension

We now use the piece of code in Figure 1 of the Farmer game again as an example for how arithmetic can be introduced with the include interface. Suppose the market price for wheat is 4 now and the price will rise 10% every iteration. Alice wants to know how much the price will be six iterations later. The rules in the Farmer game need three arithmetic functions: add, exponential, and multiply. Suppose the three functions are written in Java, C++, and Lisp, respectively. The pseudo-code of the GDL extension can be written as:

```
1.  ---------- Code written in GDL ----------
2.  include mathFromJava http://server/app/math1.jar
3.  include mathFromCpp http://server/app/math2.exe
4.  include mathFromLisp http://server/app/math3.lsp
5.
6.  use mathFromJava (java research.ggp.MathApp)
7.  mathFromJava (add ?p1 ?p2 ?p3)
8.  //call Java jar through command line parameters
9.
10. use mathFromCpp (math2.exe)
11. mathFromCpp (exponential ?p1 ?p2 ?p3)
12. //call C++ executable through command line
13.
14. use mathFromLisp (load math3.lsp)
15. mathFromLisp (multiply?p1 ?p2 ?p3)
16. //call Lisp executable through command line
17.
18. <=(inflation ?oldPrice ?rate ?step ?newPrice)
19.   (add ?rate 1 ?inflation) // inflation rate is 1+10%
20.   (exponential ?inflation ?step ?totalRate)
21.   (multiply ?oldPrice ?totalRate ?newPrice)
22.
23. //-------Code written in GGP agent for parsing--------
24. Concatenate use sentence into command line with
    known parameters assignments, e.g. concat line 6
    and line 7 into (java research.ggp.MathApp add
    10% 1 ?p3)
25. Process result list as returned by standard output
26.
27. //-----------Code written in Java-----------------------
28. package research.ggp  //java code
29. class MathApp{
30. public static void main(String[] args) {
31.   switch(args[0]) {//function selection
32.     case "add": add(args); break;
33.     case "function2": function2(args);break;
34.     case "function3":function3(args); break;}
35. }//end of main
36.
37. static void double[] transform(String[] args){
38. //parse the arguments into an array with proper type.
39.   for(int i=0; i<args.length;i++){
40.     result[index] = doubleValueOf(args[index]);}
41.   return result;
42. }
43.
44. static void function add(String[] args) {
45.   double ps[] = transform(args);
46.   out.println("%s %s %s %s", ps[0], ps[1], ps[2],
    ps[1]+ps[2]); }//send result to standard output,
    where the GGP agent collects results. (add 3 5 ?p3)
    will return (add 3 5 8)
47. }//end of class
48.
49.  //-------Code written in C++ --------
50. //omit function selection and parameter list parse
51. void exponential(char *argv[]) {
52.   double ps[] = transform(char *argv[]);
53.   cout <<ps[0]<<" "<<ps[1]<<" "<<ps[2]<<" "
    ps[1]^ps[2] << endl; }
54.
55. //-------Code written in Lisp--------
56. //omit function selection and parameter list parse
57. (defun multiply(args)
58.   (setq result (* (second args) (third args)))
59.   (remove '?p3 args)
60.   (append args (list result)))
```

Figure 2. Pseudo code for extending GDL to call external resources.

The pseudo code shows a straightforward example of how to include and use the external resources in different computer languages. The GDL code (line 1-21) is available to all players. The executables of different languages are available at some public URL for all players too. Depending on the language that the GGP agents use, the agent needs to add additional keyword parsing ability to parse include and use, just as it parses other keywords. The GDL code needs to be parsed into a command line to call the public executables with parameters (line 24). The parsing process assigns values to known parameters, and concatenates them into a command line to call the target executables. The parsing schema is independent of the language in which the GGP agent is written, or the language in which the executable is written. The agent collects standard output as the results returned (line 25). When calling, not all parameters need to be instantiated (e.g. line 24). The undecided parameters' value will be filled by the external calculation. When there are multiple assignments of the parameters, the external functions can be called multiple times, each time with one assignment list.

Instead of defining the numbers, the number sequence, and number addition as in Figure 1, the GDL add function is not defined in the game rules. Instead, the GDL uses the *include*

keyword to refer to a foreign resource and pass three parameters (two instantiated) with the *use* keyword. The same calculation that took O(n^2) in Figure 1 can be done in O(1) time now.

The foreign resource can be a shared file, a URL, or a database library. It is accessible by all players. The foreign data structure can be a list of facts, functions, or calculations. We have tested Java, C++ and Lisp. The same experiments can easily be done in other programming languages as well. The area of the specific knowledge can be game related such as resource utilization or more generally, eCommerce or a social scenario.

Note that the GGP agent does not need a corresponding compiler to invoke the executables. Only the running environment for the executables needs to be set up, just like .exe on Windows, or bash on Linux. Additional steps need to be taken for cross platform implementation such as data transfer from Windows to Linux, or to Mac, etc. In the future, the executable invoking process can be generalized into a more universal solution such as using web services or cloud computing.

To summarize, as long as the parameters can be passed back and forth through the information façade, the game functions can perform the query to complete the logic resolution. Where the data are stored and how the data are organized are not the GDL's concern any more.

3.2 Impact of the GDL Extension

The modifications and extensions of the GDL broaden the GGP area and raise the set of potential applications to a new level. The GDL is sufficiently expressive to model many relations. However, the fact that GDL is capable of describing many real world scenarios does not mean it can be done in a concise or effective manner and it certainly does not help or force the game designer to do so in a reasonable manner. Section 2.2 is an example of a tedious definition of a small group of natural numbers and addition operations. When it comes to the game goal determination or infinite real number calculation, the existing GDL structure is hindering rather than enabling new games be brought to the general game playing field. Since there exist arithmetic libraries described in other computer programming languages, redefining everything again in GDL not only adds little values to the GGP system, but also is discouraging people from turning to the GGP system for problem solving. The *include* and *use* keyword would introduce the arithmetic library and thereby make additional games possible in the GGP. Coalition games are often useful to better simulate and solve real world problems in financial, social, or industrial scenarios. With the new keywords *include* and *use* to open the interface with other languages, existing information or data from other domains can be easily hooked to the GGP engine.

4. GDL Extension Application

The external information provided via the *include* and *use* keywords is used only for the knowledge resolution and reasoning process within the game description. Different game agents are given the same game to play; therefore the same external functions to use. The logic resolution performance will be improved in a way that the external library changed "how the game is written" and will benefit all players. But the decision making and agents' performance are still completely determined by the game playing algorithms.

At the beginning of the section 3, we mentioned that the GDL extension would benefit market trading economic and incomplete information games. Coalition games are another important area of game playing research that would benefit from the GDL extension.

4.1 Coalition Games

A Coalition Game with transferable utility is defined as (N, v), where N is finite set of players; $v : 2^N \rightarrow R$ associates with each subset $S \subseteq N$ a real-valued payoff v(S) [Leyton-Brown and Shoham, 2008]. The inherent features of payoff calculations in coalition games essentially forced the GDL extension.

There have been multi-player games that require playing in groups in the GGP area. The six-player 3D Tic-Tac-Toe [see Tic-Tac-Toe 3D 6-player] requires players to play in groups of three. Each player picks the coordinate value in one dimension. The three dimensions picked by the three players determine the final position of the piece. The four player Othello [see Othello 4-player] game is similar. Four players are divided in groups of two, with each player picking one coordinate. Unlike these two games that are played in groups, coalition games require that each agent can play the games both by itself and in groups. There have not been any published coalition games in GGP.

The extension of the GDL makes describing coalition games possible. Using the extended GDL, we can explore more realistic economic and societal problems. The focus of this section is not a study of the coalition games as such but an example of the use of the GDL extension in coalition games. The results show our extensions generated reasonable results for an eCommerce coalition game.

4.2 Coalition Farmer Game

The current Farmer game is defined as a win-or-lose game. The player that ended up with the most money at the end of the game wins with 100 points awarded. If the player owns the same amount of money as another player, or less money than another player, he loses with zero points awarded.

However, the nature of the Farmer game is indeed an eCommerce game with financial achievements. In a more realistic scenario, when three players started with the same amount of money and face the same market rules, they do not have to become the richest among the three to be rewarded.

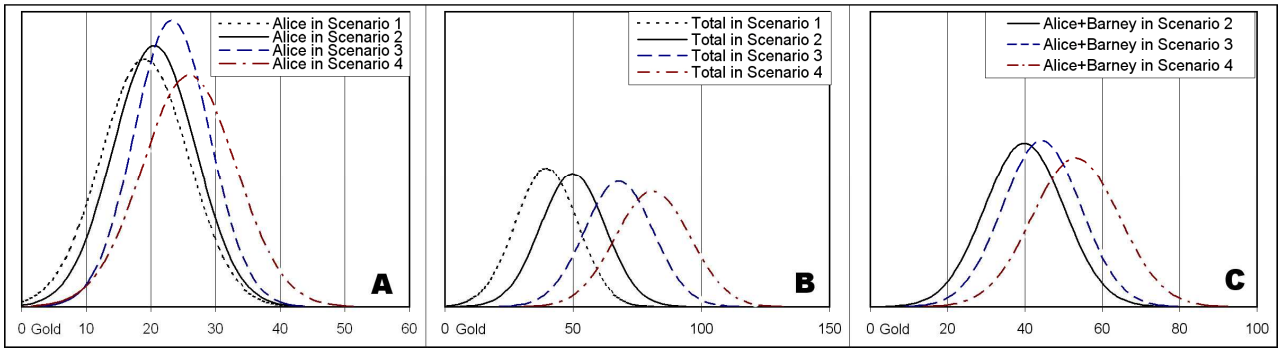


Figure 3. The normal distribution of the amount of the money the player made in different scenarios, based on 1000 matches each. (A) is gold pieces earned by player Alice; (B) is the total gold pieces of all three players; (C) is the collective results of player Alice and Barney.

The amount of money the player owns is the real reward. How much the player earned during the process matters. The player would be happier with 100 gold pieces and finishing in second place, than with 60 gold pieces and finishing in the first place. The metrics of winning is different in reality.

In addition to the amount of gold pieces earned, the three players can be more cooperative than competitive. In the open market setting with a fixed inflation number, buying a small amount in each step, holding till the end of the game, then selling together at the last step can be an easy solution for all three players to get rich. This is because pushing the price high every step with a purchase action benefits all players. In a coalition scenario like this, the goal is not beating the other players anymore, but to maximize the net worth of the entire coalition, without severely punishing any individual players of course.

The Farmer game can be rewritten into a coalition game with the GDL extension, where the performance metrics can be the total money earned by the three players together. The current inflation factor in the farmer game is set to 1 gold every iteration for simplicity. If the step-wise inflation is re-defined as 10% of the current price, there exist no reasonable way in the current GDL framework to describe such game factors without forcing the proposed extension of importing external libraries. The extension is not only convenient, but also necessary to define practical, realistic, and applicable coalition games in GGP.

With the extension of the GDL, we added the gold pieces from each of the players to get the total gold pieces earned by the three players for a coalition game. When each of the three players are trying to make their own best net worth (individually and collectively), the total value of the game (number of gold pieces as presented on the X-axis in Figure 3) increases with more cooperation.

Four experimental scenarios of the Farmer game with 1000 simulations each were analyzed. In Scenario 1, an intelligent GGP agent Alice plays against random dummy players Barney and Charlie. Alice is using the machine learning algorithms as proposed in previous research [Sheng and Thuente, 2011]. Alice earns almost twice as much as Barney or Charlie which shows the value of the intelligent agent. In Scenario 2, both Alice and Barney are using intelligent decision making for actions while Charlie remains a random player. Here both Alice and Barney earn twice as much as Charlie and Alice increases her earning over her earnings in Scenario 1. Scenario 3 has all three players using intelligent agents to make strategic decisions to maximize their individual income. In Scenario 3 both Alice and Barney improve their earnings over Scenarios 1 and 2 where they were the only one or two, respectively, trying to optimize their earnings. Scenario 4 has Alice and Barney in a coalition playing against the intelligent player Charlie. Compared to the Scenario 3, Alice and Barney both focus on maximizing the sum of their earnings. It turned out by working in coalition, the individual player's earning, the coalition earning, and the three players' total earning all increased.

The obvious observation is that when the individual players get better, the total income of the group increases (Figure 3B). In addition, if we single out Alice's income in the three scenarios, it is notable that as the other players are getting more and more capable, the income of Alice increases instead of decreases (Figure 3A). Such observations verify that it is possible to pursue the player's own interest without hurting other players in the Farmer game and therefore the Farmer game is more cooperative than competitive. We showed that, for this game, maximizing individual agent's payoff actually improves the total payoff.

To make it a more truly coalition game, we rewrite the Farmer so that "winning" requires the collective net value must be at least some number of gold pieces. We have learned from previously playing of the game in the Scenario 2 with the intelligent Alice and Barney versus random Charlie that the average collective income of Alice and Barney is 40 gold pieces (Figure 3C). So if the total of Alice and Barney is above the average 40, it is considered a coalition win for both players. In the new coalition game design, when we apply the same feature identification algorithm as in the previous research [Sheng and Thuente, 2010c], a different set

of features and weights were identified to guide the game tree search. For example, the action group *sell inventory together* is identified as important, because the price drops after each selling action so the inventory is devalued by another player's selling action. When Alice and Barney are cooperating in Scenario 4, the coalition income exceeded the addition of the individual income in Scenario 3.

To summarize, the GDL extension enables complex payoff distributions and therefore allows the new coalition version of the Farmer game, in which each of the three or more players can have their own strategy or a group strategy. The players can play individually to determine how much they can win, or in coalition with the definition of winning for the game being the collective winnings of any given coalition. Scenario 4 in Figure 3C gives the results of the coalition of Alice and Barney where they are actively cooperating to maximize the sum of their winnings. Scenario 4 provides the highest winnings for any of the examined renditions of the 10 iteration Farmer game. The collective achievements do not necessarily conflict with individual payoffs. Depending on the games rules, there exist situations where maximizing collective payoffs would strengthen individual payoffs in turn and vice versa.

5 Discussion

Since the emergence of the GGP area, the focus of the research has been on how to build a good agent. Yet this paper discusses how to incorporate more GGP games. The GDL extension opens the door for extending the GGP into new, realistic, coalition eCommerce applications. The coalition Farmer game is an example of using the modifications and extensions of the GDL. The coalition games broaden the GGP research area as we study how the dynamics change from competitive to coalition games. Fresh insights will be gained in both the specific game and GGP.

In one GGP game, the logic knowledge can be divided into two kinds. There is static information that does not change from the beginning to the end of the game (e.g. the board adjacency) and dynamic information that changes from step to step. With the GDL extension, theoretically, only game elements such as the game state, player's role, goals, state transition, legal actions, etc. need to be defined in GDL and all other computational intensive functions can be imported from or defined in user's favorite languages. However, the time spent passing parameters back and forth is considered game time cost as well. Most static information can be defined outside with the expectation of improving performance since the parameters are only passed once. The arithmetic calculations that have extremely high cost within GDL should be defined outside GDL. The minor or moderate calculation that demand frequent changes as the game state changes need to handled with caution. It is up to the game writers' judgment to determine if keeping the function definition in or outside the GDL is more efficient.

It is possible that certain GGP agents in the field today will be affected by the extension and lose their advantage in the competition. In fact, the GGP competition does not have to implement the proposed extension, as long as the researchers do not mind writing every new game with GDL for the competition. But the GDL extension proposed in this paper allows people to contribute to the GGP area without being experts in GDL. Therefore, it would give agents access to more games outside of GDL and hence attract more people into the GGP area in the long run. As GGP agents become more powerful and with this GDL extension, GGP agents may contribute to solutions in diverse domains.

The purpose of extending GDL goes beyond number calculation, even beyond introducing coalition games. It is the first step of setting up a system in which an existing game written in other programming languages does not need to be completely rewritten into the GDL to let the GGP agent play. We believe as long as the basic game elements such as initial state, player's role, legality, state transition, and goals are re-defined in GDL, a new game can be added to the GGP family without importing the data relation, computation and storage. This is one direction of our future research.

Acknowledgement

The authors gratefully acknowledge the assistance of Arpan Chakraborty concerning questions on the Lisp interface in section 3.1.

References

[Clune, 2007] James Clune. Heuristic Evaluation Functions for General Game Playing. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, pages 1134-1139, Vancouver, Canada, 2007.

[Finnsson and Bjornsson, 2008] Hilmar Finnsson and Yngvi Bjornsson. Simulation-Based Approach to General Game Playing. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, pages 259–264, Chicago, 2008.

[Genesereth and Fikes, 1992] Michael R. Genesereth and Richard E. Fikes. *Knowledge Interchange Format Version 3.0 Reference Manual*. Logic Group Report Logic-92-1, Computer Science Department, Stanford University, Stanford, CA, 1992.

[Kuhlmann and Stone, 2007] Gregory Kuhlmann and Peter Stone. Graph-Based Domain Mapping for Transfer Learning in General Games. In *Proceedings of the Eighteenth European Conference on Machine Learning*, Warsaw, Poland, 2007.

[Leyton-Brown and Shoham, 2008] Kevin Leyton-Brown and Yova Shoham. Essentials of Game Theory. Morgan and Claypool Publishers, 2008.

[Love et al., 2005] Nathaniel Love, Timothy Hinrichs, David Haley, Eric Schkufza, Michael R. Genesereth. *General Game Playing: Game Description Language Specification*. Tech-

nical report, Computer Science Department, Stanford University, Stanford, CA, 2005.
[Mehat and Cazenave, 2011] Jean Méhat and Tristan Cazenave. A Parallel General Game Player. *KI - Künstliche Intelligenz*, 25(1): 43-47, Springer, 2011.
[Möller et al., 2011] Maximilian Möller, Marius Schneider, MartinWegner, Torsten Schaub. Centurio, a General Game Player: Parallel, Java- and ASP-based. *KI - Künstliche Intelligenz*, 25(1): 17-24, Springer, 2011.
[Othello 4-player] http://euklid.inf.tu-dresden.de:8180/ggpserver/public/view_game.jsp?name=othello-fourway
[Quinlan, 1993] John Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, 1993.
[Schiffel and Thielscher, 2009] Stephan Schiffel and Michael Thielscher. A Multiagent Semantics for the Game Description Language. In *Proceedings of the International Conference on Agents and Artificial Intelligence*, Porto, Portugal, 2009.
[Schiffel and Thielscher, 2007] Stephan Schiffel and Michael Thielscher. Fluxplayer: A Successful General Game Player. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, pages 1191–1196, Vancouver, Canada, 2007.
[Sheng and Thuente, 2011] Xinxin Sheng and David Thuente. Decision Tree Learning in General Game Playing, In *Proceedings of Artificial Intelligence and Applications*, pages 192-199, Innsbruck, Austria, 2011.
[Sheng and Thuente, 2010a] Xinxin Sheng and David Thuente. Using Decision Trees for State Evaluation in General Game Playing. *KI - Künstliche Intelligenz*, 25(1): 53-56, Springer, 2010.
[Sheng and Thuente, 2010b] Xinxin Sheng and David Thuente. Using Hash Tables to Expedite Knowledge Reasoning in the General Game Playing Agent. In *Proceedings of International Conference on Advanced Topics in Artificial Intelligence*, pages 91-96, Thailand, 2010.
[Sheng and Thuente, 2010c] Xinxin Sheng and David Thuente. Predicative Sub-goal Analysis in a General Game Playing Agent. In *Proceedings of International Conference on Web Intelligence and Intelligent Agent Technology*, pages 423-427, Toronto, Canada, 2010.
[Thielscher and Zhang, 2009] Michael Thielscher and Dongmo Zhang. From General Game Descriptions to a Market Specification Language for General Trading Agents. *Workshop on Trading Agent Design and Analysis*, 2009.
[Thielscher, 2010] Michael Thielscher. GDL II. *KI - Künstliche Intelligenz*, 25(1): 63-66, Springer, 2011.
[Tic-Tac-Toe 3D 6-player] http://euklid.inf.tu-dresden.de:8180/ggpserver/public/view_game.jsp?name=tictactoe_3d_6player

A Web-Based Controlled System for Autonomous Agent

Ali Abu El Humos, Tisha Brown, Marvin Watts and Kimani Price
Department of Computer Science, Jackson State University, 1400 Lynch Street, Jackson, MS, U.S.A.
{ali.a.humos, tisha.brown, marvin.d.watts, kimani.price}@jsums.edu

***Abstract*—The technology used to control environments and devices remotely has recently seen considerable developments and improvements. In today's society, the use of these technologies is visible in government, commercial markets and academia. However, there are platforms available that need more exploration to find an economically efficient control system. The system developed to accomplish this was a web-based controlled system using an average web browser that controls a device over the Internet. There were three goals for this research: establishing a connection between the online interface and a microcontroller, determining the best microcontroller for the job and providing a user - friendly site for access. This system provides a remote user with ability to login to an embedded web server that communicates with and controls a robot .The embedded web server hosts the web pages and files that translate the commands. The robot is a commercial off the shelf autonomous robot from Parallax, Inc. The integration of these software and hardware technologies to create a cost efficient web-based controlled robot was the objective of this research.**

I. INTRODUCTION

AS of today, humans live in an age of unparalleled technological resources where embedded systems play a major role. Wireless Internet, cell phones, GPS navigational systems and computerized software are only some of the technological resources brought to us through pain staking years of research. The term embedded system is often applied to general purpose computers and also in the field of scientific investigation to the theory and practical application of AI [3]. Companies such as ATMEL, Freescale and Maxim have developed cheap and easy ways to use embedded microcontrollers which are used in many embedded systems [7].

In September of 2007, the Freescale Corporation introduced a new line to its products, called FLEXIS [7]. This moderately priced microcontroller retails for $99 dollars and provides a consumer with a low cost development system. The microcontroller uses AAA batteries and allows the user to upgrade computational requirements if needed. Many companies are introducing moderately priced microcontrollers because there is a need in the market for economically sound low-end development environments for higher – end applications. The use of the Internet allows the embedded system to offer substantial online capabilities. Web-based controlled system microcontrollers are bridging the gap between distributed sensors and actuators on one side and the Internet on the other side [2]. They are low cost devices which allow us to use standard network protocols like TCP/IP and HTTP [2].

Here is an example of how this technology can be used. Take for instance a person who has a garden in a small greenhouse. The temperature and humidity of this greenhouse must be continually monitored. The system proposed in this paper will provide the following for this individual:

1. The ability to monitor the greenhouse's environment from a remote location.
2. Provide remote access to a device through the internet.
3. Improve usability with a comfortable, easy to use web page based graphical user interface (GUI).

II. RELATED WORK

In the past thirty years there have been more technological advances than in centuries past. Contrary to popular belief, robots have proven to be an asset to the modern world. For example Industrial Robots have been a tremendous production and economic asset. Robots such as these and others have changed the landscape of society not only industrially, but academically, financially, socially and even politically. Current revolutions in computer and embedded design have given rise to the exploration of remotely controlled devices through the use of an embedded web-based system. In [2], the authors present research that focus on networked embedded greenhouse monitoring and control based on simple embedded web servers and 1-wire protocol for connecting sensors and actuators. The hardware and software architecture of embedded web servers are described and the experimental results of monitoring and control are presented. The embedded web server is described as a low-cost and efficient microprocessor that facilitates the use of the Internet. An embedded web server is ideal in this research with a commercial off the shelf robot In [5], the authors outlined a wireless sensor network for monitoring a habitat. The emphasis of the research focused on the design of the sensor network, nodes, and a remote access database manager. The requirements for the system include Internet accessibility, assigned hierarchical networks, sensor longevity, archiving system behavior and sampling management. Providing the user with Internet access and simple interfaces creates compatibility and integration to countless other applications and devices. The researchers independently designed the board for their

research. Many researchers take this path because there is a lack of resources available to allow individual users to control, monitor and access devices on their terms. The issues are the reliability of these small low capacity microcontroller , communication via the Internet and web server functionality.

The basic function of an embedded system is to acquire and accumulate data about the status of the object, and to control its operation [1]. The importance of exploring this topic can aid in improving the capabilities of many consumer products that are used daily and improving the creation of a platform to guide the developers of web-based controlled systems.

III. SYSTEM ARCHITECTURE

There have been many viewpoints discussed on establishing a criteria or standards for communicating through the Internet by way of an embedded web server. The objective of this research is to develop a system where a remote user can login through a webpage and communicate with a robot using an embedded web server. The embedded web server will host the web pages and will run an embedded version of Linux. The Parallaxtm Boe-Bottm using an interface, provided online, is what is used for this research.

The NGW100 is a network gateway that provides web applications and has a 4MHz Atmel AP700 microcontroller with a 512KB storage unit. The board is capable of mounting extra storage and extension headers. NGW100 uses a 9V power supply and is very energy efficient. The user interface for the system is stored on the NGW flash drive. The board runs a version of Linux, interfaced with the 802.11 network to send data to the local devices or drives.

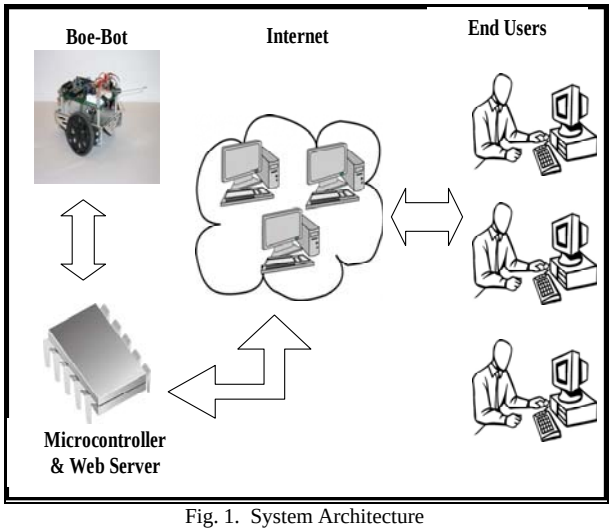


Fig. 1. System Architecture

The NGW100 Board is implemented to function as the gateway communication between the Parallax Boe-Bot and the Internet. After communication is made, the webpage or "Web-Based Controlled System" serves as the graphical user interface (GUI) to send directional commands serially to the Boe-Bot using common gateway interface (CGI) scripts. The embedded server will provide a suitable interface to network the Stamp Module used by the Boe-Bot to the Internet.

IV. METHODOLOGY

The following outline details the procedure used to implement the web-based controlled system and test its operability.

1. Build and test the functionality of the Boe-Bot as a stand alone module.
2. Setup the embedded web-server and create the necessary web pages for user interaction and test its functionality as a stand alone module.
3. Establish a communication link between the Boe-Bot and the embedded web-server.
4. Test the functionality of the overall system.

Since the Boe-Bot is an off the shelf robot its setup is straight forward. However, it will take a considerable amount of time and effort to configure the NGW100 for internet access. Several items needed to setup a NGW100 board are a serial cable, a COM port on your PC or desktop, a terminal window such as HyperTerminal or minicom, Ethernet cable, a 9v power supply and an SD card. The following is a step-by-step guide for configuring the NGW100.

A. IP Configuration

a. Setup terminal window
b. Set PC IP configuration
c. Disable eth0 on NGW100
d. Reserve IP address with udhcpc

The IP configuration is a very crucial part of the NGW100 configuration for this experiment. This step allows the user to communicate to the NGW100 web server through various different web browsers. This also enables communication to other vital elements such as the general input/output pins (GPIO) device controller.

B. Build Root Environment

a. Download development environment
b. Build new uImage for NGW100
c. Load uImage
d. Test environment

Buildroot is a set of scripts and menu system which builds an entire root file system for a given target. In this case the target is the NGW100. After compiling and executing Buildroot a development environment is available for the user to implement and create customizable applications or programs for the NGW100 board. In this environment, a

user can build new Linux images, modules and firmware to modify the NGW100 board.

C. *Build GNU Tool Chain*

a. Build GNU Tool Chain
b. Compile C test program
c. Transfer executable of test program to NGW100
d. Run test

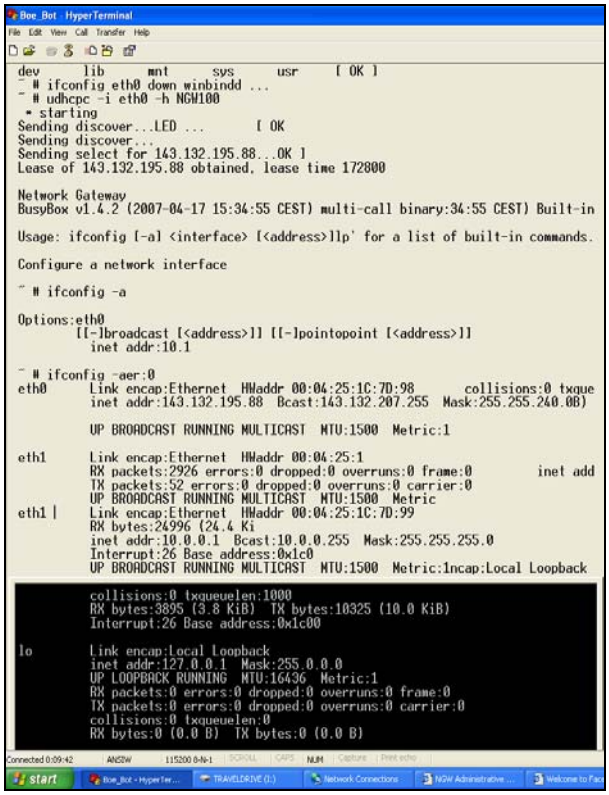


Fig. 2. NGW100 Data HyperTerminal

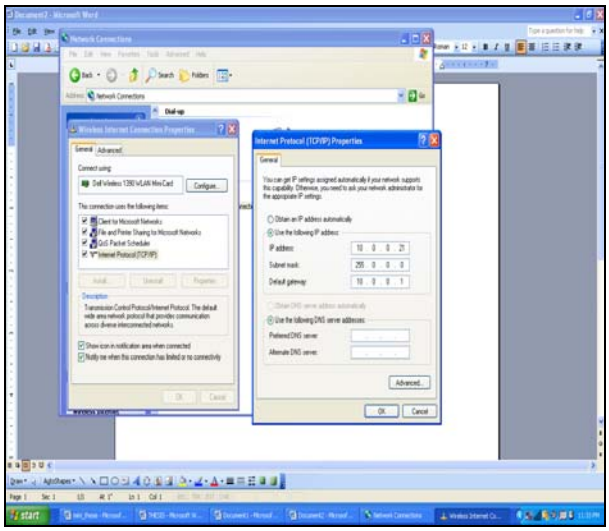


Fig. 3. PC IP - Configuration

GNU ToolChain is a collection of developmental tools. It includes tools such as a compiler, assembler, linker and binutils, source code libraries, programmer and debugger. These applications allow the user to define their specific environments and construct the features needed in the applications using code. Cygwin was chosen as the platform for creating the system because of its affordability, industry performance and many features, like performance analysis and design tools for detailed applications. Several trial and error scenarios were performed in the process of implementing code.

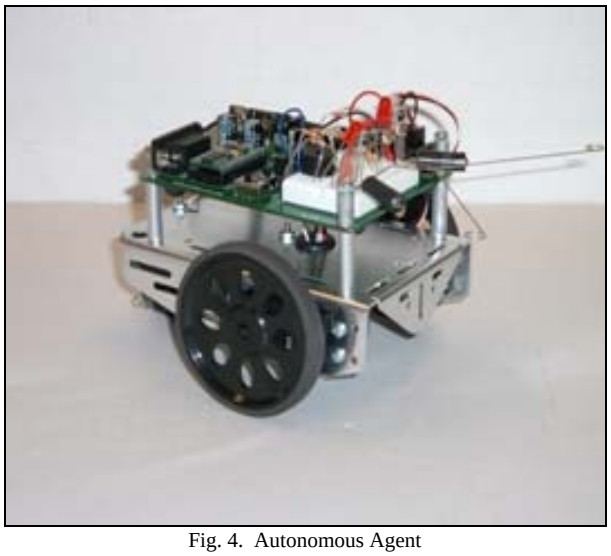


Fig. 4. Autonomous Agent

V. COMMUNICATION

During the experiment it became necessary to enable extra serial ports to facilitate PC to serial communication. These extra ports were enabled by building a new uImage for the embedded Linux operating system installed, and transferring that image to the board. The ports were then enabled by updating the following C Language code in the setup.c file:

```
at32_map_usart(0, 1);   /*USART 0 to /dev/ttyS1 */
at32_map_usart(2, 2);   /*USART 2 to /dev/ttyS2 */
at32_map_usart(3, 3);   /*USART3  to /dev/ttyS3 */

at32_add_device_usart(1); /* USART1 device */
at32_add_device_usart(2); /* USART2 device */
at32_add_device_usart(3); /* USART3 device */
```

Communication is handled by commands sent from an HTML web page to a CGI file on the server that interprets those commands. From this point the NGW100 device can control the autonomous agent directly. The practical and simple implementation of this system will be very useful in future system designs of this type.

Figure 5 shows the Web-based controlled system that an individual uses to send commands through hypertext transfer protocol (HTTP) to the server connected directly to the autonomous agent (Robot). The system is a simple web

browser on the user's machine independent of any other software installations. Once the request is received the CGI script is executed. The CGI file translates the HTML code into the necessary commands that the autonomous agent can execute. The system allows the user to engage in a step-by-step selection of commands. After clicking on the command, the step response page generates a result for your application and sends it to the device. The system is unique because of its ability to combine all aspects of a web-based controlled system.

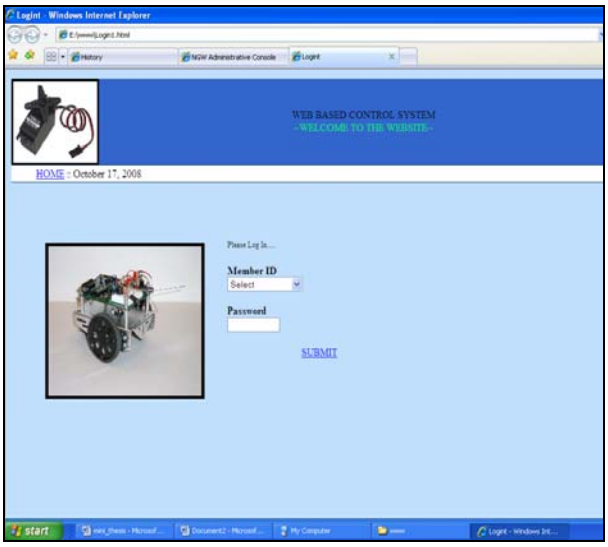


Fig. 5. Web-based controlled system

VI. CONCLUSIONS AND FUTURE WORK

The web-based controlled system explained in this paper introduced the concept of reducing the cost and complexity of the web–based system to increase the versatility and range of user availability. Users that once lacked the tools and guidance can now reap the positive benefits of this system. The low complexity of this design makes this system suitable for the average in home user who would like the luxury of remote monitoring.

After deciding on the microcontroller, the specifications of the system were decided. The device used to test the functionality of the control system was the Parallax Boe-Bot. Important additions to the systems include firmware and software up-grades. Development of web-pages was a critical part of the process because they provide the interface to control the autonomous agent. CGI files were used to define predetermined functions.

Once all of these components were added, the testing phases begin. The performance of the system was measured on the ease of operation, user friendliness, and efficiency. The successful implementation of the web-based controlled system highlights the capabilities of low cost microcontrollers and research ingenuity.

Future work on the system will include the addition of cameras mounted on the autonomous agent as well as wireless communication between the autonomous agent and the web server.

REFERENCES

[1] J.Teng, C.Tseng and Y.Chen,, "Integration of Networked Embedded Systems into Power Equipment Remote Control and Monitoring" , Department of Electrical Engineering, I-Shou University, Kaohsiung, Taiwan 2004.

[2] D. Stipanicev, D. and J. Marasovic,, "Networked Embedded Greenhouse Monitoring and Control", *Control Applications IEEE Conference* Volume 2, 23-25 June 2003 Page(s):413-420.

[3] J. Hill and D. Culler, "A Wireless Platform for Deeply Embedded Networks", *Mica: IEEE Micro*, 22(6):12-24, November/December 2002.

[4] M. Horton, D. Culler, K. PIster, J. Hill, R. Szewczyk, and A. Woo, "MICA, The Commercialization of Microsensor Motes", *Sensors*, 19(4):40-48, April 2002.

[5] A. Mainwaring, J. Polastre, R. Szewczyk, and D. Culle , "Wireless Sensor Networks for Habitat Monitoring", *ACM International Workshop on Wireless Sensor Networks and Applications*, September 2002.

[6] A. Abu-El Humos, B. Alhalabi, M. K. Hamza, E.Shufro, and W. Awada , "Remote Labs Environments (RLE): A Constructivist Online Experiment in Science, Engineering and Information Technology", *The 31st Annual Conference of the IEEE Industrial Electronics Society, IECON05*. Raleigh, North Carolina, USA 6-10 Nov, 2005.

[7] "Microprocessor Report, Reed Electronics Group" Available from: *http://www.freescale.com/files/microcontrollers/doc/support_info/FLEXIS_ARTICLE1.pdf*.

[8] "A Brief History of Robotics". Robotics.MegaGiant.com History. Available from: http*://robotics.megagiant.com/history.html*.

[9] M. Heidar, "Web-Based Control Systems Laboratories using LabView", *Journal of Engineering Technology*; Spring 2003.

[10] G. Ingvar, "A Remote Laboratory for Electrical Experiments", *Proceedings of the American Society for Engineering Education Annual Conference & Exposition*; Spring 2002.

Economic Effects of Multiple Intelligent Agents in Online Auctions

Jacob Sow[1], **Patricia Anthony**[2], **and Chong Mun Ho**[3]
[1,2] School of Engineering and Information Technology, Universiti Malaysia Sabah, Kota Kinabalu, Sabah, Malaysia
[3] School of Science and Technology, Universiti Malaysia Sabah, Kota Kinabalu, Sabah, Malaysia
[1]jacob07.my@gmail.com, [2]panthony@ums.edu.my, [3]cmho@ums.edu.my

Abstract - *Online auction mechanism has enriched the trading community. It brings together both sellers and buyers from around the world without the geographical limitations and time constraints. In this paper, an auction market is simulated with multiple English auctions that are running concurrently consisting of human bidders and intelligent agents with various bidding strategies. For the analysis, we considered the market economy such as the auction closing price trend and the number of auctions that were closed successfully. We also studied a situation where a market is fully populated by different types of intelligent agents. Based on the results obtained, we discovered that bidders received more surplus and they are more satisfied when they are using bidder agents rather than bidding in the auctions themselves. From the seller's perspective, they may not welcome these bidder agents since their revenues are reduced in a market with finite number of auctions and participants.*

Keywords: English auctions, market economy, intelligent agents, bidding strategies.

1 Introduction

When considering the agents mediated electronic marketplace, agents play an active role in both sellers and buyers sides. A seller agent may advertise its products in the market, stating the selling price and looking for the potential buyers in the market. On the other hand, a buyer agent would look for the desired goods or services requested by its user and it has a task to bargain about the price of the products and find the best deal [1]. Both parties that may be located in different parts of the world need not gather at a specific location to perform trading. Therefore, sellers are now looking for a larger group of potential buyers to buy their goods, while buyers are looking for a better offer for their desired goods in the online marketplace. There are basically four different types of auction protocols namely the English auction protocol, the Dutch auction protocol, the first-price sealed bid and the second-price sealed bid auction protocols that are widely applied to trade a single item. Regardless of which protocol is used in an auction, bidders may face the difficulty of finding a suitable auction to participate since there are hundreds of thousands of online auctions available on the Internet. Thus, agent technology is introduced with the hope of easing this problem. In online auctions, computer agents can be deployed as search agents, monitoring agents or even bidding agents on behalf of their human users. Furthermore, intelligent agents are suitable to be deployed in online auctions because auctions follow certain protocols which are well defined and procedures are clearly stated [1].

There are many researches performed on different bidding strategies which can be applied to these agents. However, literatures on the online auction market economy are relatively few. Among these literatures, some of them are focused on the human bidders or sellers with different risk behaviors and their preferences in auctioning and bidding processes such as [2], [3], [4], [5]. As the agent technology in auction marketplace is becoming a more dominant trend, it would be interesting to explore the market economy when agents are implemented. For example, the economic effect in a market that is fully populated by intelligent agents is one of the interesting topics to be discussed. In [6], Lee and Malmendier found that human bidders tend to overbid in the auction participated. This would indicate that sellers obtain higher revenues when such scenarios are found in their auctions. When homogeneous or heterogeneous intelligent agents are found in an auction market, sellers may assume their revenues may be lowered since these bidder agents do not overbid and always make wider survey such as the available auctions than human bidders before making decision on the auction to be participated and the bid amount to be submitted. In this paper, we would like to investigate the market economic effects from the perspective of bidders when they are represented by bidder agents. Moreover, sellers' reaction when bidder agents are introduced is also examined briefly. In this paper, a simulated marketplace is used to generate multiple English auctions and participants. Data from this market is collected for further analysis and discussion. In the data analysis phase, we investigate the winning utility, the average closing price of auctions won, the quantity of desired items procured and surplus ratio received by winners of various types. In the first measurement, winner's surplus is calculated with respect to his private valuation. Secondly, we calculate the average payment in auctions according to winner's types. Thirdly, the quantity of the desired items which are successfully obtained by winners of various types is considered. Lastly, the consumer surplus ratio is use to measure the surplus received by the winners. It does not only consider the surplus but takes into account the number of bids received in an auction and the median number of bids received across all auctions generated. In Section 2, literatures on the English auction protocol, risk behaviors of standard bidders and agent technology that

are implemented in online auctions are reviewed. This is followed by the discussion on the market setup in Section 3. In Section 4, the results and discussions are elaborated. Finally, in Section 5 we conclude the experiment and discuss avenues for future works.

2 Literature Review

2.1 English Auction Protocol

In [7], Hu and Bolivar showed their interest in online auction efficiency, particularly on eBay auctions. They investigated multiple online auction properties including consumer surplus and their cross-relationships. They implemented consumer surplus ratio (CSR) as a measurement to evaluate the winner's profit over their private valuations. Also, they introduced the concept of median in this CSR in order to reduce the influence of extreme bid amounts received in an auction. By comparing the CSR, they found that the surplus ratio is generally impacted by the nature of the market and the ability to find a replacement of the item.

In another research done by Lee and Malmendier [6], they examined the overbidding in online auctions. They found that overbidding happens not only in common-value model, but it does happen in private-value model auctions. They considered this phenomenon as the bidder's curse. They further explained that this phenomenon is caused by the cost of switching, the limited attention and limited memory of bidders and the extra winning utility enjoyed by winners. From their data collected, they concluded that overbidding happened in eBay auctions, and some bidders failed to recognize the alternative fixed price option. Also, they suggested that auctions that attract overbidders would in turn increase the sellers' revenue.

2.2 Bidder's Behaviors

Bierman [8] studied three distinct types of risk behaviors from the perspective of game theory. He further explained these three risk behaviors using von Neumann-Morgenstern utility function. In his explanation, a risk-averse (RA) person would receive relatively smaller marginal utility from winning a lottery compared to the marginal utility of losing the lottery. Secondly, a risk neutral (RN) person would receive the same marginal utility from winning or losing the lottery. Thirdly, a risk seeking (RS) person would receive relatively larger marginal utility from winning a lottery compared to the marginal utility of losing lottery.

Using von Neumann-Morgenstern utility as a reference, many researchers further examined the impact contributed by these bidders in different auctions. Talluri and Ryzin [2] found that a RA customer will prefer the first-price auction to the second-price auction. Besides that, in the first-price auction, RA bidders may bid more aggressively and hence result in higher sellers' expected revenue [2] [3] [4]. So, from the sellers' point of view, they may prefer first-price auction than the second-price auction when bidders are risk-averse [2] [3] [4] [5].

In their respective papers [9] [10] [11], Roth and Ockenfels mentioned that many bidders (or their agents) tend to submit their bids late or also known as sniping. In their research, they found that sniping behavior avoids three distinct bidding wars, namely bidding wars with incremental bidders, with like-minded late bidders and with uninformed bidders who look to others' bids to determine the value of an item. Finally, they concluded that the last-minute bidding is not simply due to naïve time-dependent bidding, but it responds to the strategic structure of the auction format in a predictable way.

2.3 Agent Technology in Online Auctions

According to [12] [13], an intelligent agent is a computer entity that is capable of flexible autonomous action in order to meet its design objectives. These agents should be autonomous, interactive, responsive and proactive.

Anthony and Jennings [14] developed a bidding agent for multiple heterogeneous auctions. With its bidding strategy, bidder knows how much to bid in an auction and would withdraw from a certain auction if the current bid is higher than the suggested bid. In short, in this research, they presented a novel bidding strategy that could be used across multiple auction protocols with varying start and end times.

In another study conducted by Lim *et al.* [15] [16], they argued that an intelligent agent would greatly help its bidder if the closing price of an auction is predicted successfully. In their research, they proposed the grey prediction model because the number of historical data required is minimal. In an environment such as online auctions, rapid response is necessary. Therefore, when there is insufficient historical data, the grey prediction model seems to be the most appropriate method to be used.

He *et al.* [17] [18] focused their interest on agent's bidding strategy that is incorporated with neuro-fuzzy techniques. This strategy identifies auctions that are most suited to the bidders' requirements and according to their risk attitudes, bids in some other auctions that have approximately similar expected return, but which close earlier than those in the best return set. From the results obtained, this algorithm performed better among other strategies considered. This was due to its ability in selecting the auctions to bid according to the relatively correct prediction on the closing prices of the auctions.

3 Simulated Marketplace

3.1 Simulated English Auctions Market

There are many successful online auction houses that are running on the Internet such as eBay and Amazon Auctions. However, due to different

perspectives of houses administrators and researchers, data retrieval from these auction houses is restrictive if not impossible. Therefore, simulated online auction marketplace becomes an alternative testing platform for researchers to conduct their experiments and retrieve data for further analysis.

This marketplace simulates a real auction house where multiple English auctions are conducted. All auctions in this marketplace are the symmetric independent private values (SIPV) auctions [5] 19]. It is assumed that:

1) Each auction is selling a single indivisible object (single-unit auction).
2) Bidders know their own private valuations only. Even if he knows about other bidders' valuation, this would not contribute to a change of his valuation.
3) All bidders are indistinguishable (symmetry).
4) Unknown valuations are independently and identically (iid) distributed and using continuous random variables (independence, symmetry and continuity).

In this simulated marketplace, the number of English auctions to be conducted is set in a range of 1 to 100 inclusive. Besides that, two different groups of participants are considered, namely the standard bidders and bidders who use agent technology. Their population is set in a range of 0 to 3000 inclusive. In the first type, standard bidders can be subcategorized into three distinct groups according to their risk behaviors (risk aversion, risk neutral and risk seeking). In another type, intelligent agents can be subcategorized into the Greedy agents [20], the Heuristic agents [14] and the Sniping agents [7] [9] [10] [11]. Every auction generated is assigned a reserve price randomly from a normal distribution with its mean and standard deviation collected from the real online auction data. Private valuations of standard bidders and bidder agents are also generated from the same distribution. Therefore, it simulates the real online auctions pricing scenario. After that, standard bidders are assigned to different auctions by the system. Whenever possible, standard bidders are distributed evenly to all the auctions generated. They are treated as the faithful bidders as they do not move from one auction to another. Thus, as their assigned auction starts, they would submit their bid and outbid one another provided that their private valuations are not violated. Conversely, intelligent agents do not remain in any auctions but they move around the auctions in the marketplace. At each time step, these intelligent agents would find the most promising auction to participate.

During the bidding process, for each active auction, a standard bidder will be chosen randomly to submit a bid. Meanwhile, if there is any intelligent agent interested in submitting its bid in the same auction, a competition among the selected standard bidder and the intelligent agent(s) occurs. As a result, a higher bid outbids the lower bids.

3.2 Intelligent Agents in the Simulated Market

The first type of agents used in the market is the Greedy agents. A Greedy agent would always look for an auction with the lowest current bid as its target auction (Fig. 1). Some bidders may use their agents to look for auctions with the lowest current bids. By doing so, they wish to purchase the items with minimal prices.

The second type of agents is the Sniping agents. These agents would hold their bids until the last time step of an auction with the hope of outbidding others giving them insufficient time to react (Fig. 2). Bidders who are impatient with longer auction closing time and keen to obtain the items desired without wasting time on surveying other auctions may prefer this strategy.

while ($t < t_{\max}$) *and* (item not obtained = *true*)
- Build active auctions list
  - List all auctions that are active before $t_{\max}$.
- Select potential auctions from active auctions list to bid in.
  - Select target auction as one that has the lowest current bid.
- Calculate the new bid, current bid + randomize bid increment.
- Bid in the target auction with the new bid.

End while
where
t is the current universal time across all auctions;
$t_{\max}$ is the agent's allocated bidding time by when it must obtain the goods or leave the auctions.

Fig. 1: The Top-level Algorithm for the Greedy Agent.

while ($t < t_{\max}$) *and* (item not obtained = *true*)
- Build active auctions list
  - List all auctions that are active before $t_{\max}$.
- Select potential auctions from active auctions list to bid in.
  - Select target auction that has t = end time - 1.
- Calculate the new bid, current bid + randomize bid increment.
- Bid in the target auction with the new bid.

End while
where
t is the current universal time across all auctions;
$t_{\max}$ is the agent's allocated bidding time by when it must obtain the goods or leave the auctions.

Fig. 2: The Top-level Algorithm for the Sniping Agent.

while ($t < t_{\max}$) *and* (item not obtained = *true*)

- Build active auctions list
  - List all auctions that are active before $t_{\max}$.
- Calculate the new suggested bid using the agent's strategy.
  - If the difference(suggested bid, current bid) > a preset threshold, new suggested bid = current bid + a portion of difference
- Select potential auctions from active auctions list to bid in.
  - Select target auction as one that maximizes agent's expected utility.
- Bid in the target auction with the new suggested bid.

End while

where

t is the current universal time across all auctions;

$t_{\max}$ is the agent's allocated bidding time by when it must obtain the goods or leave the auctions.

Fig. 3: The Top-level Algorithm for the Heuristic Agent.

Lastly, the Heuristic agents take into account the remaining time, the available auctions, the desire for a bargain and the desperateness of obtaining the items. By combining these tactics, a new suggested bid is produced (Fig. 3). This type of agents would be implemented by bidders who are well prepared before participating in any auction. To bid in a given auction, they consider the current bid, the number of similar auctions available, the timeline of obtaining the items if won and how desperate they are in procuring the items.

4 Experimental Analysis

In this experiment, heterogeneous standard bidders and heterogeneous intelligent agents are generated in the same marketplace. Some expectations listed below are examined:

1) When more agents are found in the marketplace, the auction closing price is reduced.
2) Agents perform better when there is no standard bidder located in an auction.
3) Sellers may not prefer intelligent agents in their auctions compared to standard bidders.

4.1 Experimental Setup

In this experiment, there are six different bidders' types and the competition occurs not only between standard bidders and intelligent agents, but also within the groups of standard bidders and intelligent agents with different types. 90 auctions and 900 bidders (both standard bidders and intelligent agents) are generated. 5 situations are considered in this experiment (Table I). The same marketplace is repeated 10 times and the data are collected for data analysis. From this market, the performances of the bidders are evaluated by using the average winner's utility, the number of winning auctions, the average closing price and the consumer surplus ratio (CSR). In addition, the reaction of sellers when they are confronted with multiple types of bidders and agents is also studied.

The winner's utility is measured as

$$U(v) = \left(\frac{p_r - v}{p_r}\right) + c, \qquad (1)$$

where v is the winning bid, p_r is the winner's private valuation and c is an arbitrary constant 0.001 to ensure that the winner receives some rewards when he wins by paying his private valuation. Each winner's utility is calculated and lastly they are summed and averaged. This measurement is used to calculate the intrinsic reward a winner receives by winning an auction. Furthermore, this intrinsic value is calculated with respect to his private valuation to reduce the influence of different private valuations of different winners.

TABLE I. PROPORTION OF PARTICIPANTS IN A MARKETPLACE

Situation	**0**	**1**	**2**	**3**	**4**
Standard Bidders	900	810	540	270	0
Intelligent Agents	0	90	360	630	900

Next, the average numbers of auctions won by each group of standard bidders and agents are counted as

$$\overline{W_i} = \frac{\sum_1^n W_i}{n}, \qquad (2)$$

where $\overline{W_i}$ is the average number of auctions won by winners of type i, W_i is the number of auctions won by winners of type i in each run and n is the number of runs conducted in this experiment. This method is used to evaluate the number of winning auctions in a society of a certain type of winners.

Thirdly, the average closing prices paid by different groups of bidders/ agents are also recorded. These values are calculated as

$$\overline{C_i} = \frac{\sum_{j=1}^{n_i} C_{ij}}{n_i}, \qquad (3)$$

where i represents different groups of winners (standard bidders or intelligent agents), n_i indicates the number of auctions won by winners of type i and C_{ij} represents the

closing price of auction j submitted by winners of group i.

Lastly, the CSR is measured as

$$CSR_j = median_{\forall j}\left(\frac{\left(v_{H_i} - v_{F_i}\right)\cdot\left(N_i + N_m\right)}{v_{F_i}\cdot N_i + v_{H_i}\cdot N_m}\right), \quad (4)$$

where j is the type of winners, v_{H_i} is the winner's private valuation in auction i, v_{F_i} is the final winning bid in auction i, N_i is the number of bids of item i and N_m is the median number of bids across all the auctions conducted. This ratio is used to show the surplus ratio gained by each winner with respect to his private valuation. It considers the number of bids received in each auction and the median number of bids received across all the auctions available. Hence, the factor of number of bids is minimized as in various auctions; some auctions may only receive a small number of bids compared to others which in turn would affect the closing price of that auction.

4.2 Results and Discussions

Fig. 4 shows the average winner's utility obtained in this experiment. In all the situations analyzed, Greedy and Heuristic agents obtained higher winner's utility compared to other standard bidders. It is much obvious in a market with the absence of the standard bidders (Situation 4). In other words, the Greedy agents and Heuristic agents not only completed their tasks assigned, they successfully achieved higher satisfaction of their bidders with higher intrinsic values. It can be credited to their bidding strategies which would lead them to suitable auctions and suggest an appropriate bid to be submitted. Conversely, the Sniping agents scored the lowest utility in all the situations because of its attitude of always searching and sniping in the auctions that would close in the next time step. As a result, they ended up paying higher prices to obtain the items.

Next, the average number of winning auctions obtained by different groups of winners is illustrated in Fig. 5. As more agents are located in the market, the number of auctions won by standard bidders decreases. From the standard bidders' point of view, as more agents of different types are found in the marketplace, standard bidders with RS behavior may help them in achieving higher number of winning auctions even though the difference may not be significance among the standard bidders community. This is observed by comparing the number of auctions won by three distinct types of standard bidders from Situation 0 to Situation 3. On the other hand, in terms of the number of winning, Sniping agents successfully outperformed the other two types of agents in Situation 2 where one third of the agent population was using the sniping strategy. In this case, Sniping agents are useful in obtaining items desired when their population is relatively smaller in the market. Nonetheless, Heuristic agents performed best in Situation 3 with 20.6 auctions won on average. In the absence of standard bidders (Situation 4), the Heuristic agents showed their excellent performance by winning more than 50% of the auctions. In all the situations analyzed, the overall closing numbers are 89.6, 89.4, 89.6, 89.5 and 87.5 respectively out of 90 auctions generated. These numbers may not be equal to 90 as some of the auctions are closed without winners.

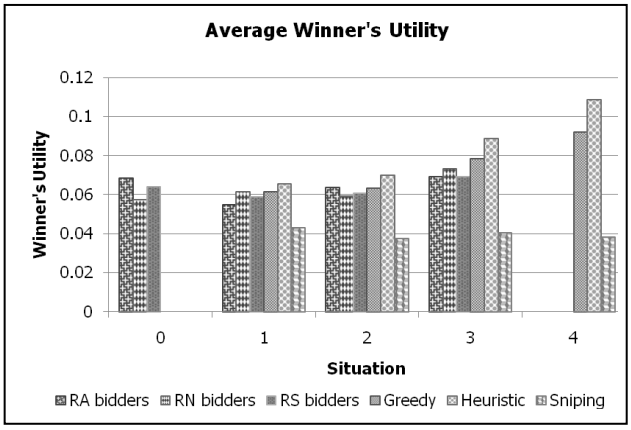


Fig. 4: Average Winner's Utility according to Winner's Types.

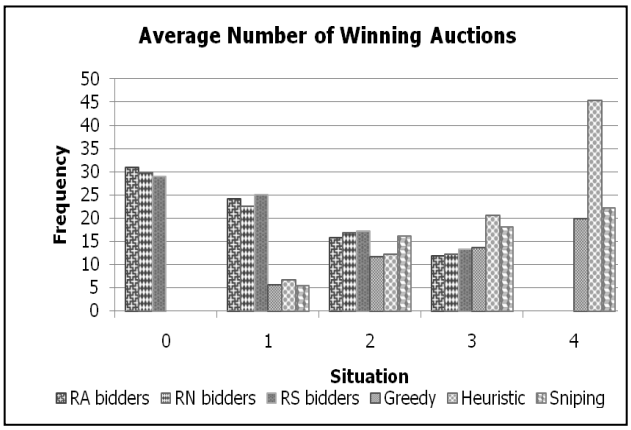


Fig. 5: Average Number of Winning Auctions according to Winner's Types.

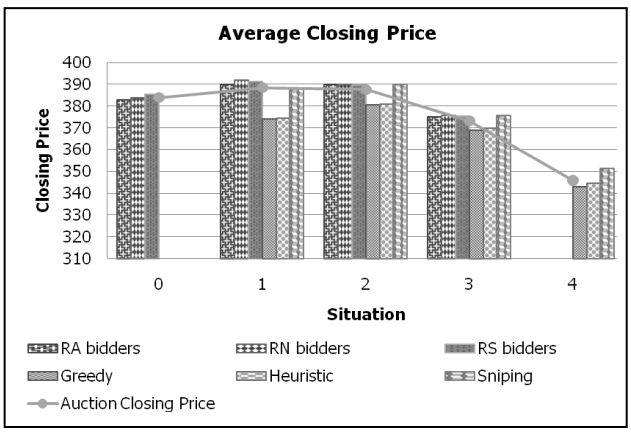


Fig. 6: Average Closing Price according to Winner's Types.

The average closing price is depicted in Fig. 6. Generally, as more agents participate in an auction market, the closing price decreases. This can be seen from Situation 1 to Situation 4. Agents with their superior bidding strategies, successfully avoid bidding wars with standard bidders in the market. However, there is an increase in the closing price from Situation 0 to Situation 1. When more bidders' types are found in the market, it affects the competition that occurs within the market. Besides that, when comparison is made among the intelligent agents, it can be seen that Sniping agents pay more in obtaining the items desired. Their winning bids are greatly affected by the winning bids of the other standard bidders. They always snipe in auctions at the very last possible moments. In this experiment, this moment is referred to as the last time step before an auction's end time. Thus, they have to submit higher bids to overbid the current leading bidders and it causes their closing prices to be the highest among the agents community. In Situation 4, it can be observed that even though the Sniping agents still achieved the highest closing price in the agent population, their price was greatly reduced from Situation 3 (375.84) to Situation 4 (351.31). It may be due to the absence of standard bidders who faithfully bid in their auctions assigned and eventually drive higher closing prices. The involvement of intelligent agents reduces the final closing prices of the auctions they participated and therefore brings more satisfaction to the winners. However, from another perspective, sellers' revenues are decreasing as a direct result from this reduced closing price.

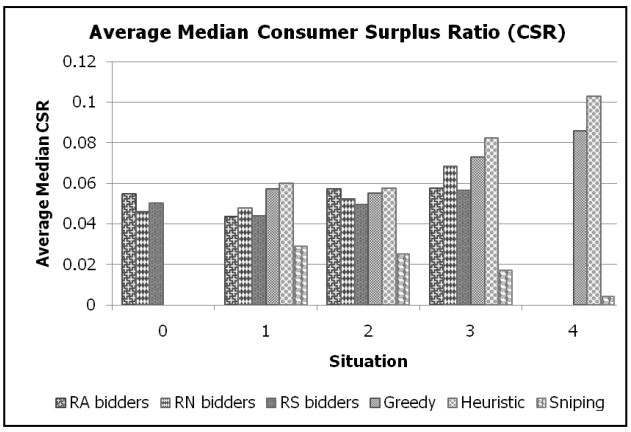


Fig. 7: Average Median CSR according to Winner's Types.

In Fig. 7, the average median consumer surplus ratio (CSR) of different types of winners is illustrated. It can be seen that as more agents are situated in the marketplace, they obtain higher CSR than the standard bidders. However, there is an exception where the CSR of Sniping agents is worsened. Its surplus ratio is further reduced to 0.00435 in Situation 4. It indicates that in a market where no standard bidders are present, most of the Sniping agents receive 0.00435 as their surplus ratio if they win the auctions. Therefore, it is shown that by using Sniping agents in this market, bidders hardly receive larger surplus especially when the populations of another two types of agents (Greedy and Heuristic agents) are growing.

Based on the results of this experiment, the auctions closing prices decrease as more bidders are represented by their bidder agents. The decreasing trend becomes more apparent in a market where only agents are present. Secondly, when there is no standard bidders present in the market, intelligent agents perform well especially the Heuristic agents. On the other hand, from the sellers' perspective, they may not welcome bidder agents compared to standard bidders. This is supported by the total number of auctions closed with winners and the auction closing price. The number of auctions that closed with transaction is reduced from 89.6 (Situation 0) to 87.5 (Situation 4), roughly 2.34% negative growth rate. Besides that, the auction closing price decreases from 375.84 to 351.31 at a rate of approximately 6.53%. Consequently, in this market with finite number of auctions and participants, sellers may not be satisfied by using bidder agents since the implementation of intelligent agents would affect their profits gained from the auctions.

5 Conclusion and Suggestion

In conclusion, as expected, intelligent agents outperformed their opponents in all the measurements used. Their excellent performances are clearly observed in a market where standard bidders are absent. However, there is an exception among the agents population. The Sniping agents did not perform well in obtaining desired goods and did not achieve greater saving compared to the other two types of agents. However, to those bidders who are not concerned with the surplus and are desperate in obtaining the item required, the Sniping agents may become one of their options since by using these Sniping agents, they would obtain the goods desired within a short period of time and the price paid is always within their private valuations. Generally, from all the situations considered in this experiment, the auction closing prices are decreasing as more intelligent agents are present in the market. Moreover, these agents successfully help their bidders to obtain the items desired while trying to increase their satisfaction by achieving greater saving. In addition, the number of auctions that eventually completed with transactions is reduced when no standard bidders are found in the market. Thus, in a market where numbers of auctions and participants are finite, sellers may prefer more standard bidders in their auctions than intelligent agents as more standard bidders may raise their revenues gained from the auctions.

In the future, we plan to investigate further this bidding behavior in terms of market economy. In this paper, standard bidders of different risk types are assigned to different auctions and they are not allowed to move from one auction to another. In the real auction market, this type of faithful bidders is hard to find. A more realistic approach would be that they are faithfully submitting their bids until the auction closes, if they do

not win, they may participate in other active auctions until their constraints are met (such as the time or private valuation).

Moreover, all the bidding strategies implemented in this paper do not include the past historical data of the items being auctioned and therefore are not able to predict the auction closing prices. As suggested in [15] [16], it would be useful if these intelligent agents are equipped with prediction capability in participating auctions. With the predicted closing price, agents can make better decision based on their private valuation and these predicted prices.

6 References

[1] Dignum, F. 2001. Agents, Markets, Institutions and Protocols. In Dignum F. and Sierra C. (eds). Agent Mediated Elec. Commerce, LNAI 1991, pp. 98-114. Berlin: Springer-Verlag.

[2] Talluri, K. T. and Ryzin, G. J. V. 2004. Auctions. In Talluri, K. T. and Ryzin, G. J. V. (eds.). The Theory and Practice of Revenue Management 69, pp.241 – 297. New York: Springer.

[3] Klemperer, P. 1999. Auction Theory: A Guide to the Literature. Journal of Economic Survey. 13: 227 – 286.

[4] McAfee, R.P. and McMillan, J. 1987. Auctions and Bidding. Journal of Economic Literature. 25(2): 699-738.

[5] Wolfstetter, E. 1999. Auctions. In Wolfstetter E. (ed). Topics in Microeconomics: Industrial Organization, Auctions and Incentives, pp. 182 – 242. Cambridge: Cambridge University Press.

[6] Lee, Y. H. and Malmendier, U. 2007. The Bidder's Curse. Working Paper 13699. Cambridge: National Bureau of Economic Research.

[7] Hu, W. & Bolivar, A. 2008. Online Auctions Efficiency: A Survey of eBay Auctions. Proceedings of the WWW 2008, Beijing: ACM. 925 – 933.

[8] Bierman, H. S. 1998. Uncertainty and Expected Utility. In Bierman, H. S. and Fernandez, L. (eds.). Game Theory with Economic Applications, pp. 207 – 238. Boston: Addison-Wesley Publishing Company, Inc.

[9] Ockenfels, A. and Roth, A. E. 2002. The Timing of Bids in Internet Auctions: Market Design, Bidder Behavior, and Artificial Agents. Journal Artificial Intelligent Magazine. 23(3): 79 – 88.

[10] Roth, A. E. and Ockenfels, A. 2002. Last-Minute Bidding and the Rules for Ending Second-Price Auctions: Evidence from eBay and Amazon Auctions on the Internet. American Economic Review. 92(4): 1093 – 1103.

[11] Ockenfels, A. and Roth, A. E. 2006. Late and Multiple Bidding in Second Price Internet Auctions: Theory and Evidence Concerning Different Rules for Ending An Auction. Games and Economic Behavior. 55: 297 – 320.

[12] Jennings, N. R. and Wooldridge, M. 1998. Application of Intelligent Agents. In Nicholas R. Jennings and Michael Wooldridge (eds.). Agent Technology: Foundations, Applications and Markets, pp. 3-28. Berlin: Springer-Verlag.

[13] Wooldridge, M. and Jennings, N. R. 1995. Intelligent Agents: Theory and Practice. The Knowledge Engineering Review. 10(2): 115 – 152.

[14] Anthony, P. and Jennings, N. R. 2003. Developing a Bidding Agent for Multiple Heterogeneous Auctions. AMC Transactions on Internet Technology. 3(3): 185 – 217.

[15] Lim, D., Anthony, P. & Ho, C. M. 2007. Evaluating the Accuracy of Grey System Theory against Time Series in Predicting Online Auction Closing Price. Proceedings of the 2007 IEEE International Conference on Grey Systems and Intelligent Services, November 18 – 20, 2007, Nanjing, China.

[16] Lim, D., Anthony, P., Ho, C. M. & Ng, K. W. 2008. Assessing the Accuracy of Grey System Theory against Artificial Neural Network in Predicting Online Auction Closing Price. Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 IMECS 2008, March 19 – 21, 2008, Hong Kong, China.

[17] He, M., Jennings, N. R. & Prugel-Bennett, A. 2004. An Adaptive Bidding Agent for Multiple English Auctions: A Neuro-Fuzzy Approach. Proceedings of the IEEE International Conference on Fuzzy Systems, July 25 – 29, 2004, Budapest,Hungary. 1519 – 1524.

[18] He, M., Jennings, N. R. and Prugel-Bennett, A. 2006. A Heuristic Bidding Strategy for Buying Multiple Goods in Multiple English Auctions. ACM Transactions on Internet Technology. 6(4): 465 – 496.

[19] Matthews, S. A. 1995. A Technical Primer on Auction Theory I: Independent Private Values. Discussion Paper No. 1096. Evanston: Northwestern University.

[20] Byde, A. 2002. A Comparison among Bidding Algorithms for Multiple Auctions. In Padget, J.,Shehory, O., Parkes, D., Sadeh, N. and Wlash, W. E. (eds). Agent-Mediated Electronic Commerce 2002, LNAI 2531, pp.1 - 16. Berlin: Springer-Verlag.

A case study of task-based reorganization in a pursuit game simulation

Maryamossadat N. Mahani [1], **and Arvin Agah**[2]
[1]EECS Department, University of Kansas, Lawrence, KS, USA
[2]EECS Department, University of Kansas, Lawrence, KS, USA

Abstract— *Important to the effective performance of a multi-agent system, is using the appropriate organizational structure. Different organizations might be appropriate for different tasks, goals, and environmental conditions. As a result, sometimes a higher performance can be achieved by reorganizing to a more promising structure. One of the problems in reorganization is defining the criteria upon which to evaluate and find a promising organization to reorganize to. Considering the costs that can be associated to reorganization, most of the research to-date has been focused on reorganizing when system performance falls below a certain threshold. While some opportunities for reorganization might be lost. In this work, we perform a simulation study to identify effects of a task-based model of reorganization. The results demonstrate performance enhancements using our approach despite the costs associated to reorganization.*

Keywords: Multi-agent systems, Organizational structures, Reorganization, Pursuit game simulation

1. Introduction

Autonomic systems, capable of adaptive behavior are envisioned as a solution for maintaining computing systems that are situated in dynamic environments. The uncertainties and limitations in perceptual, computational, and communication abilities of these systems create the need for cooperation. Different assignment of roles, authority and communication relations between the system components affects the overall performance [10]. Many systems employ an additional layer of structuring known as organizational structure to control these assignments. The structure of an agent organization significantly affects performance characteristics of a system [5]. As a result, reorganizing to a more appropriate structure can be beneficial or essential when the environment or system conditions change.

One of the problems in reorganization is defining the criteria upon which to evaluate and find a more promising organization to reorganize to. Different approaches taken by the research community to initiate a reorganization include setting a performance threshold [12], finding inefficient patterns of communication [14], setting a time limit, or a certain number of simulation steps [12]. These approaches can fail to identify all cases that a reorganization can be beneficial.

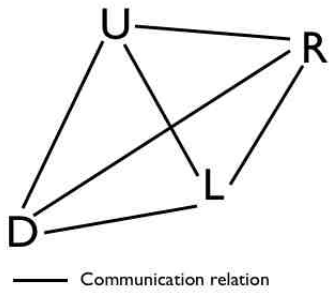
At the same time, to date relatively a small amount of work has been done on reorganizing to a different model of organizational structure. Most systems reorganize within the same structural model [15], [11], [14], [16]. One example of such reorganization would be restructuring the agents in a hierarchical model of organization by changing the depth or width of the hierarchy.

The aim of this work is to use simulations to explore performance effects of reorganization to a different model of organizational structure using a task-based approach for triggering the reorganization. With this purpose, a reorganization is initiated upon changes in task type and based on a heuristic model which creates a mapping between task-types and organization structures. We use a pursuit game simulation to create a platform and verify a preliminary model of a task-based reorganization.

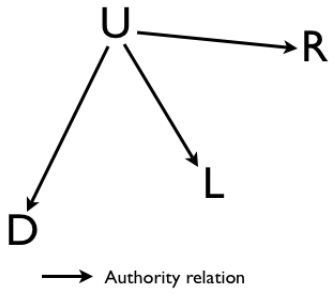
2. The Organization Model

We consider organization as an already existing and predefined structure that the agents need to conform to. In this work, we employ a limited model of a problem solving agent organization with the following specifications:

- Each agent has a specific skill.
- The agents are invariant and their properties do not change over time.
- Each agent can become the member of one group at a time.
- Tasks change over time.
- Each task requires a combination of different agents with complementary skills to be accomplished.
- Agents are only self-aware (they have no global view of the world).
- Agents can form acquaintance, communication, or authority relations, with each relation spanning the properties of the previous relation in this list. Agent relations change over time.
- Agents have no relations with each other at the beginning. Once an agent joins a group, then it forms a relation with other group members. The agent relations in each group depend on the group type and the agent role in that group.



(a) Communication relations in a coalition



(b) Authority relations in a team

Fig. 1: Agent relations in different organizations

We employed S-MOISE+ [13] as the organizational modeling framework. This framework enables us to make sure that the agents within the system follow the organizational constraints. Using this framework we have implemented coalitions and teams as two different models of organizational structure. We have also implemented a reorganization model which allows structural change between coalition and team structures. A short description of each of these organizational models follows.

- Coalition: We consider coalition as a goal-directed and short-lived group of agents that is formed with a goal in mind and dissolves as soon as the goal is satisfied [9]. All agents in a coalition are peers and have a communication relation. The agent that initiates formation of a coalition, acts as the representative for the coalition. Figure 1.a demonstrates communication relations of four agents that are part of a group formed as a coalition.
- Team: A team is considered as a number of cooperative agents that coordinate to be supportive of the team's goals [9]. In this work, each team will have a leader that maintains an authority relation with all the other team members. Figure 1.b demonstrates agent relations in a team with agent U as the leader.
- Reorganization: We consider reorganization as a structural change [8] between different organizational models in which the structural elements such as agent roles or authorities change. We employ an organization-centered model for controlling the reorganization [12].

3. Use of simulations in verifying effects of task-based reorganization

The pursuit game has been a popular problem for addressing cooperative behavior in multi-agent systems [6]. In this work we create the world as a 20 by 20 grid with boundaries but no obstacles. The world gets populated with a certain number of hunters and preys. Figure 2. presents a screen capture of the world populated with 8 hunters and 2 preys. The pursuit simulation has the following specifications,

- All agents move with the same speed.
- All agents can sense objects within a radius of 8 around them.
- All agents can only move in vertical and horizontal directions.
- The world gets populated with agents randomly distributed on the grid.
- A hunter moves randomly if it has not detected any prey itself nor it is receiving any information from its group members.
- Each hunter will have a certain skill from four possible skills. An Up-hunter tries to position itself above the prey that it is trying to catch, a Down-hunter tries to position itself below, Right-hunter to the right, and Left-hunter to the left.
- Each prey tries to maximize its distance from the hunters that it can sense. A prey moves randomly if it does not sense any hunters.
- A prey is caught and removed from the world when it can not move any more. A prey will not be able to move if it is surrounded by hunters or by a combination of hunters and the world boundaries.

3.1 Cooperation and task decomposition in a Pursuit Game

The goal of hunters is catching all preys in the world. As the hunters only have access to local information, it becomes necessary that they cooperate and share information. Also, as each hunter has a specific skill, cooperation of hunters with complementary skills becomes necessary for catching a prey. In order to catch a prey, the hunters need to perform two sequential tasks. The first task is detecting a prey. Once a prey is detected, the second task is tracking the prey to catch it. An accurate detection of a prey requires a combination of at least 2 to 4 hunters sensing the prey at the same time. When a hunter senses a prey, it tries to form a group with other hunters. Once a group is formed, agents in the group can share information about preys. A well-formed group of hunters will have one skill from each of the 4 possible skills. No group can have more than one agent from each type of skill. The organizational framework ensures that group formation follows the specified requirements.

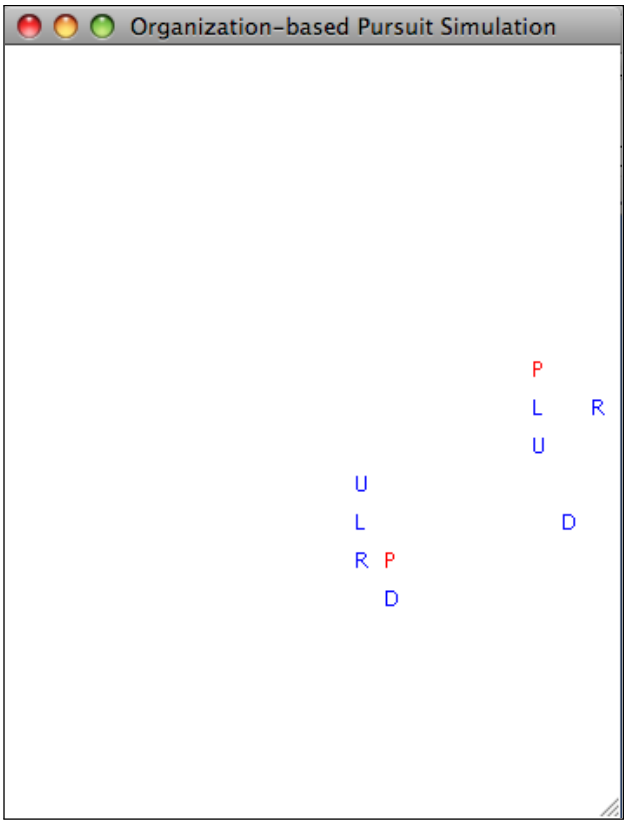


Fig. 2: Screen capture of a sample pursuit game simulation

A coalition structure might be more time-effective for detecting a prey as all hunters will be able to communicate and share information. At the same time, it might be more cost-effictive to have one leader agent to keep track of the prey and share that information with the rest of group members.

3.2 Experimental Evaluation

We have run three sets of simulation runs with three models of organization and reorganization: 1. Only coalition, 2. Only team, 3. Reorganization between coalition and team. In a coalition model, a coalition structure is used for both detecting and tracking a prey. In this model, each agent relies on its own sensory information if it senses a prey, and shares that information with the rest of group members if it is part of a group. If it does not sense a prey and it is part of a group, then it looks to use any shared information. If no shared information is available either, then it moves randomly. Similar to a coalition model, in a team model a team structure is used to detect and track preys. With the difference that only the agent that leads the team can share information with the rest of group members. Also, if a team member has received information from the team lead, then it will just use the shared information instead of using its own sensory information.

In a reorganization model, coalition is used to detect a prey and team is used to track and catch a prey. Once a coalition is formed and a prey is detected, then a reorganization is performed by changing the roles and authorities to those of a team. The coalition representative agent will become the team lead, and the rest of coalition members will become team members.

System performance is measured in terms of resource usage and timeliness of goal achievement. The cost associated to each agent in the organization is measured both in terms of number of messages and number of moves that the agent has to make. The total communication cost is measured as

$$CommCost_{ORG} = C.\sum_{x=1}^{A} c_x \quad (1)$$

where C is the communication cost coefficient, and c_x is the number of messages sent by agent x in that time-step. c_x will also include any messages passed for reorganization. The total move cost is measured as

$$MoveCost_{ORG} = M.\sum_{x=1}^{A} m_x \quad (2)$$

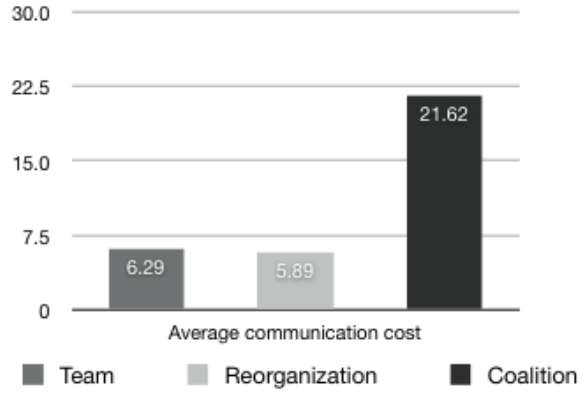
where m_x is the number of moves for that agent. The total cost is a combined measure of the total communication and move costs.

$$TotalCost_{ORG} = CommCost_{ORG} + MoveCost_{ORG} \quad (3)$$

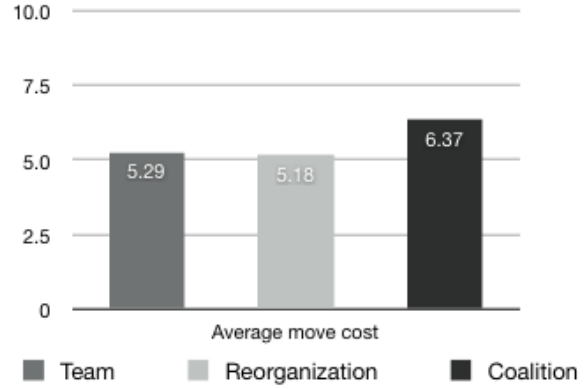
We conducted the experiments by running 50 simulation runs for each organization model in a pursuit game populated with 4 hunters and 1 prey. We present the results in terms of average cost and time. The graphs in Figures 3.a, 3.b, and 3.c show an overview of the average communication cost, move cost, and the average time for each organization model. The results indicate that the reorganization model outperforms the static models despite the extra communication costs associated to reorganization. The results also demonstrate the difference that various models of static organization can make on performance. In this case, the static coalition structure introduces a significantly higher communication cost compared to the static team structure. This re-emphasizes the importance and effects of using an appropriate organizational structure.

4. Conclusions and Future Work

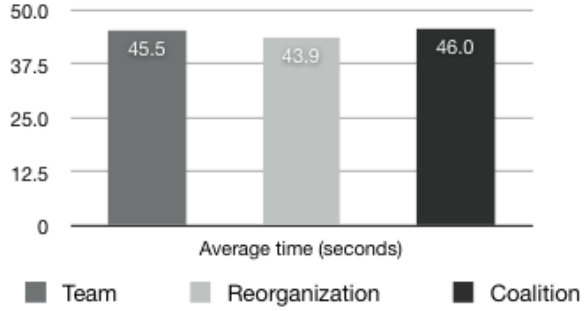
This paper addresses the problem of reorganization in problem solving agent organizations. More specifically, we address reorganization between different models of organizational structure by means of a task-based method for triggering the reorganization. We apply a heuristic model of reorganization according to which a structural change is enforced by the organization in order to switch to a more efficient model of agent cooperation for each task



(a) Average communication cost



(b) Average move cost



(c) Average time

Fig. 3: Average time and organization costs

type. Thus, this work demonstrates a simple and robust method for increasing the efficiency of problem solving agent organizations.

We are interested in relations between task types and the effective organizational structure for performing that task. Future work will initially involve creating a framework for modeling task types to include cooperation and goal achievement needs for a task. We will be getting insights from social organization theory to create a knowledge base of heuristics for selecting an appropriate organizational model for each task type. We will enhance the model by adding a learning module so the system can learn from past experiences. We also intend to expand our experiments into other types of problem solving agent organizations and also different scales of agent population. Another stream of future work will focus on verifying effects of the frequency of task type changes on the overall system performance when employing this reorganization method.

References

[1] Simon Fear. (2005). Publication quality tables in LaTeX. [Online]. Available: http://www.ctan.org/tex-archive/macros/latex/contrib/booktabs/booktabs.pdf

[2] Leslie Lamport. *"LaTeX: A Document Preparation System."* Addison-Wesley Publishing Company, 1986.

[3] Wikibooks contributors. (2008) LaTeX Wikibooks, collection of open-content textbooks. [Online]. Available: http://en.wikibooks.org/wiki/LaTeX

[4] American Mathematical Society. (1999) User's Guide for the amsmath Package (Version 2.0). [Online]. Available: ftp://ftp.ams.org/pub/tex/doc/amsmath/amsldoc.pdf

[5] K. M. Carley and L. Gasser, Computational organization theory. In: G. Weiss, *Multiagent Systems: A Modern Approach to Distributed ArtiÞcial Intelligence*, pp. 299Ð330. MIT Press, 1999.

[6] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan and K. K. Saluja," Sensor deployment strategy for target detection," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pp. 42–48, Atlanta, Georgia, 2002.

[7] V. Dignum and F. Dignum, "Modelling Agent Societies: Coordination, Frameworks and Institutions", in *Progress in Artificial Intelligence*, LNAI 2258, pp. 191–204, Springer-Verlag, 2001.

[8] V. Dignum and F. Dignum, " Exploring congruence between organizational structure and task performance: A simulation approach", in *Coordination, Organization, Institutions and Norms in Agent Systems I*, pp. 213–230, Springer, 2006, vol. 3913 of LNAI.

[9] B. Horling, *Quantitative Organizational Modeling and Design for Multi-Agent Systems*, PhD thesis, Department of Computer Science, University of Massachusetts at Amherst, 2006.

[10] B. Horling and V. Lesser, "A Survey of Multi-Agent Organizational Paradigms", The Knowledge Engineering Review, number 4, pp. 281-316. Cambridge University Press, 2005, vol. 19.

[11] B. Horling, R. Mailler and V. Lesser, V., "A Case Study of Organizational Effects in a Distributed Sensor Network", in *Proceedings of the International Conference on Intelligent Agent Technology*, pp. 51-57, Beijing, China, 2004.

[12] J. F. Hubner, J. S. Sichman and O. Boissier, "Using the MOISE+ for a cooperative framework of MAS reorganisation", in *Proceedings of the 17th Brazilian Symposium on Artificial Intelligence*, pp. 506-515, Berlin, Springer, 2004, vol. 3171 of LNAI.

[13] J. F. Hubner, J. S. Sichman and O. Boissier, "S-moise+: A middleware for developing organised multi-agent systems", in *COIN I*, pp. 64–78, Springer, 2006, vol. 3913 of LNAI.

[14] R. Kota, N. Gibbins and N. Jennings, "Self-Organising Agent Organisations", in *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, pp. 797–804, Budapest, Hungary, 2009.

[15] M. Sims, D. Corkill and V. Lesser, "Automated Organization Design for Multi-agent Systems", in *Autonomous Agents and Multi-Agent Systems*, pp. 151-185, Springer-Netherlands, 2008, vol. 16, number 2.

[16] P. Stone and M. Veloso, "Task Decomposition and Dynamic Role Assignment for Real-Time Strategic Teamwork", in *Proceedings of the 5th International Workshop on Intelligent Agents, Agent Theories, Architectures, and Languages*, pp. 293–308. Springer-Verlag, London, UK, 1999.

Query Based Learning in Multi-Agent Systems

Safiye Sencer
Department of Business Administration, Abant Izzet Baysal University, Bolu, Turkey

Abstract - *This study focuses on query based learning in multi-agent systems which include both data management operations and coordination activities. The study is oriented on agent based and database systems with model driven approach (MDA) which provides arrangement of data within a multi-agent system by letting filter with query based learning which supports the decision mechanism within the system. It uses the similarity measure on maximum entropy approach, which is often to find out interesting and meaningful patterns from databases. At the same time, it may generate a variety of rules, such as classification rules, throughout to learning rules of the query based learning process.*

Keywords: Multi-agent system, Query based learning, Model driven approach, Entropy

1 Introduction

The query based learning system considers the multi agent system's dynamic knowledge base to realize the learning process. In particular, query based learning modeled with system behavior under dynamic conditions with MDA. Model driven approach includes many features of an object oriented programming notation: classes, associations, attributes, methods, primitives and enumerations. It contains sketch, design, analysis of testing and program models. MDA usually uses modeling programming languages which include object oriented approaches like the UML (Unified Modeling Language). The query based learning algorithm brings a new opportunity for agent learning methodology, which is how to build systems, how to obtain decision rules and how to make decisions in multi agent systems.

Some examples of Model driven approach (MDA) used with query structures in some studies are below. Usually, modeling of the main characteristics of events and functions represented in an object dimension display the interrelation attributes in more detail. Various studies consider the simple operation of the query mechanisms within multi-agent systems with a MDA (Song et al. 2006, Philippi, 2006) [1,2]. Some studies include the query operation mostly used in choosing related information in web environment such as query based access issue on web environments [3], [4], [5]. Also some Web database application studies include the query based access with data preparation and type recognition approaches with SQL with relational database within query interfaces and query schemes (author, title, subject for amazon.com) using the data mining approach [6-14]. Modeling of the intelligent query answering mechanism suggested in multi agent systems [15]. Some studies discussed query processing in agent systems with peer to peer network architecture structure [16,17]. Also, query mechanism discussed in some architectures such as distance learning[18], query agent and answer structure[19], processing and mapping theory processes given in the database within the agent structure[20], NZDIS project and agent based distributed information systems architecture[21].

For the studies carried out with the model driven approach where the agent were represented with model driven. The studies discussed the relationship between components and connector structures with a goal driven approach [22]. Dynamic features such as architectural goal structure and change replacements, interfaces and behavioral specifications were also discussed. The architecture of the autonomous agent was examined and presented related classes and interface by describing the agent structure in grid service and generally mentioned the query structure among these relationships [23], [24], [25]. Static and dynamic modeling components of the agent based information systems applied in the probabilistic model [26].

The model driven approach in the agent-based simulation applied on elements, organization, agents' environment timers, environment space, tasks, goals, organization groups, interaction, and protocol specification. Meta-model and agent behavior of the agent systems expressed in the model driven approach [27]. The model driven architecture analyzed within the ontology structure, discussed the meta model with UML structure and defined the model transformation with the model driven approach and studied on UML [28, 29, 30]. MDA and query structure analyzed agent oriented design and implementation in general via using within UML [31]. The sensor network with query processing is suggested [32]. Mostly, realized studies have considered the query structure not the query based learning mechanism used in MDA.

In this paper, Section 2 discusses agent based query process; section 3 introduces the agent-based query learning system; section 4 suggests the query-based algorithm; section 5 describes query based learning application and section 6 represents the paper's conclusion.

2 Agent Based Query Process

The distributed problem solving approach is an ideal solution for multi agent based learning system for information coordination and integration in heterogeneous, distributed and

dynamic environments. However agents' their information sources and interrelationships are highly dynamic, inadequate in their network environments. This study focuses on dynamic data management operations and coordination activities among peers and their databases with query based learning system in a multi agent system. In particularly, this paper proposes an agent-based cooperative information system architecture consisting of multi agents, which cooperate with each other to resolve information retrieval and integration problems which presents a query based learning mechanism in multi-agent systems with a model driven approach (MDA).

Query process checks the whole data for try to find goal key conditions. Briefly, the query is analyzed to extract its key phrases and query type (e.g. what, where, when, how, how much) and consider the attributes, relations and selected condition situations. Generally in SQL accomplished the Table 1 situation.

Table 1 General SQL conditions

SELECT	Attributes
FROM	Relations
WHERE	Selection Conditions

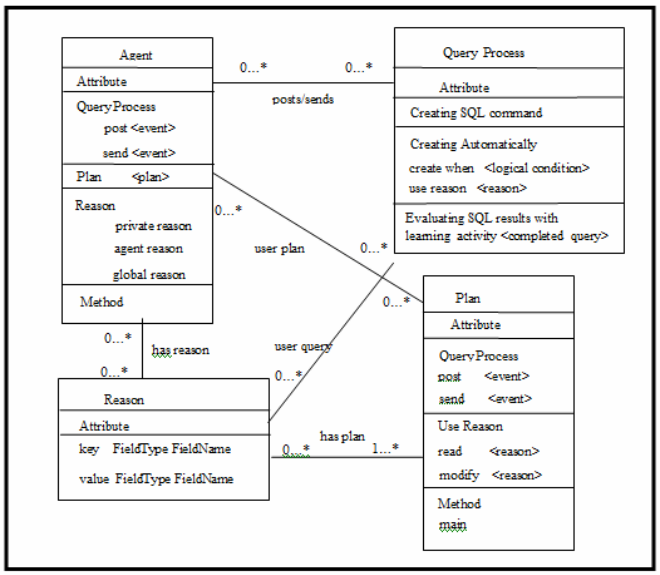


Figure 1. Structure diagram of agent based query system

Query based learning mechanism (QLM) system ensures arrangement of data within a multi-agent system by letting them filter with query process for the decision mechanism within the system. Figure 1 represents a conceptual map, which includes the correspondence to semantic associations and relationships among agent based query process, plan and belief structures, which are given in multi-agent structure. Each class includes an attribute and data. The relations between each other functional structure is shown in UML. The main characteristic of the system is a learning mechanism, which is able to evaluate and interpret the automatic SQL query especially the query process, has realized within this structure and the related results. Each task is considered as being an autonomous entity corresponding to a goal or to a sub-goal in hierarchical structure. The agent class realizes query process with support class such as reason and plan. It gives detail information about system structure.

3 Agent Based Query Learning System

Query based learning is a part of machine learning which optimizes the performance criterion using current and past situation data. The model defined up to some parameters, and learning is the execution and optimizes the parameters of the model using the past experience. The suggested model able descriptive to gain knowledge from data, derivative to obtain rule from knowledgebase with query then predict future decisions. When interface realizes the query process, consider the following requirements. In order to identify and use the characteristics relevant to the task to be taken into outline in the interface level components are: information and resources; control parameters and activations. Information and resources includes detail information about task, cases, attributes with name, goal, index and hierarchy framework. The dynamic knowledge base presents with input and output interface. Control parameters aim to check and detect some failures in the system with temporal constraints, error toleration and functional limits. During the realization of the query based learning, system realizes some activation such as querying, reasoning, collaborating, planning and acting (Fig. 2).

Suggested learning mechanism used system's static and dynamic variables. Especially, dynamic variables are continuously changeable. If system adapt dynamic situation easily, it will reflect to system ability. Learning activity used the query-plan and reason activity process with system resource and control variables.

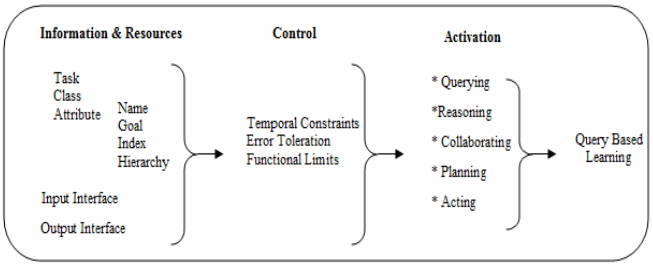


Figure 2. Interaction of learning activity

Information gain is measured and evaluated by entropy. The system's feature has the highest gain in entropy, which is chosen as the first splitting feature. This method is robust with regard to noisy data and is capable of learning discrete valued functions. However, it does not handle continuous-valued

function and missing values. Moreover, an over fitting problem does not exist for this algorithm. The suggested system learns the knowledge obtained as a result of query as a rule or task. The system fulfils not only the task but also the learning process. Learning process is acquired and the data from the external transition is processed by the agent system of the defined aim during the activities.

Query based learning mechanism consist of knowledge processing, query processing and query optimization. In the whole multi agent system, some situations as an agent, interaction and environment, which will be, affect the system's learning situation. Here, things that facilitate the factor's perceiving of the environment can be listed as predictability, accessibility, dynamics, variability and the current states of the resources. Especially agent numbers, abilities and goals are related to the system's learning ability directly. Also, environmental factors that dynamic, diversity and resource perceived the resource from environment.

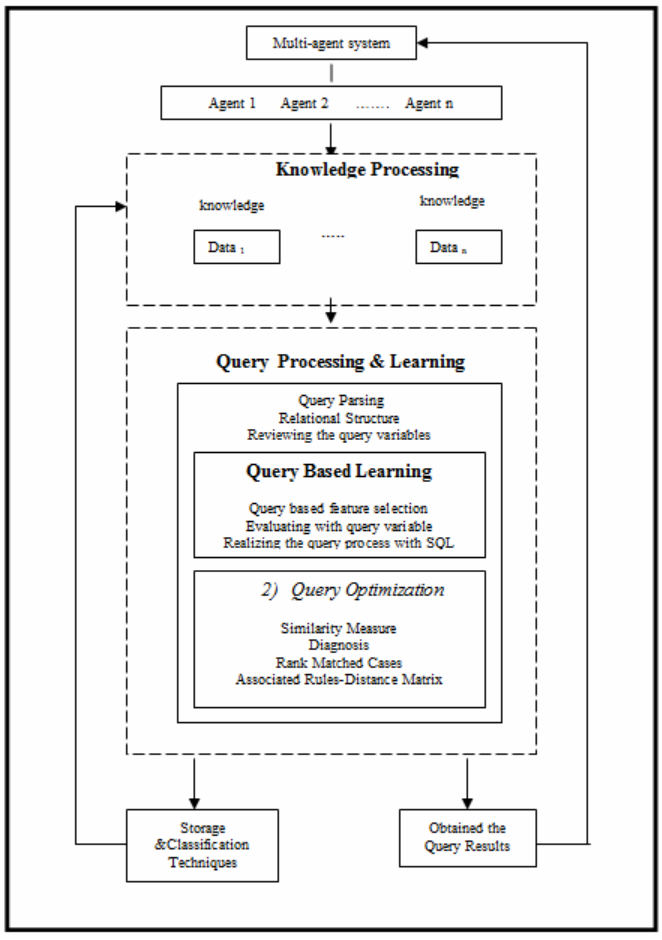


Figure 3. Query Based Learning Mechanism

The operation of Query Learning Mechanism is given in Figure 3. The figure explains the database of each agent, the Query operation applied to these databases and operation of query learning mechanism in connection with query optimization and database as a result of this operation. This mechanism creates the rule knowledge to be used by the system through combining the information obtained from database, query optimization and realizing the query process with SQL.

Query based learning mechanism consist of four main step. These are knowledge processing, query processing, query based learning and query optimization.

3.1 Knowledge Processing

Knowledge gathering, storage and classification process is realized in this stage. The criteria and keywords to be used in evaluating the received data are defined in this stage. This stage can also be called pre-query. The keywords, concepts, attribute and relationship knowledge to be analyzed by the agent are determined in this stage before query.

Therefore, the followings are fulfilled in knowledge processing:

- Goal definition
- Data selection
- Data preparation

knowledge can be included in the query process in evaluation from the beginning of the problem understanding to the end when the result inferred by the predictive model is presented to the users while used in practice. It is necessary to understand the project objectives and requirements and then convert them into a data mining problem definition. In the proposed process model QBLS, a domain knowledge base is used especially for the results post-processing and for the missing input values pre-processing in the Use of the Model phase. Some features included in the final model may not be directly provided by users but can be inferred by the domain knowledge base.

In general the external domain knowledge base assists to deal with the vague queries in use of the model phase and with eliminating the illogical outcomes in post processing.

3.2 Query Processing

Query processing includes the query parsing, relational structure and reviewing the query variables. Also it is called as a preparation stage for query based learning. Query parser process translates an input query into a stream of events represented as integers and the XML elements symbolized as knowledgebase. At the same time, the query parser checks the validity of the query and then translates it into an internal form usually a relational calculus expression, or related item equivalent. Relational database uses the set of mathematical terms and SQL terminology, which are roughly equivalent to SQL database terminology. The query variable process behavior reviews in the Table 2.

Table 2 the SELECT statement

```
SELECT    [distinct] attribute_names
FROM      table_list
[WHERE     conditions ]
[GROUP BY colum_list
[HAVING conditions ]]
[ORDER BY column_list [DESCENDING OR ASCENDING]]
```

3.3 Query Based Learning

The agent performs two types of query in the process of defining keywords, concepts or attributes during knowledge processing. The first is external query, which is realized among the agents, while the second is the internal query, where the agent scans the knowledge within itself. During these query processes, the SQL approach is applied.

3.4 Query Optimization

Query optimization consists of similarity measure, diagnosis, rank matched cases and associated rules distance matrix processes. Obtained final results classified and made storage then sent to agent knowledge and rule base system. Then the query results are sent to main system then system evaluate for the next decision position. Query based learning and optimization explained in Section 4 in detail.

4 Query Based Learning Algorithm

In this research in the dynamic agent based query based learning structure aimed with interactions among the agents to research in the multi agent system and cooperation and particularly the negotiation during the making decision with interactions. In learning algorithm entropy based weight and feature selection preferred and evaluated in the suggested system. When query based feature selection realized actively, query-learning process could be efficient. Query based feature selection is developed for the numeric prediction task as well as the classification prediction task. Query feature selection is compared and evaluated the query keyword and database attributes.

In particularly, entropy used for feature selection and query based learning.

$$H(P) = -\sum_{i} p_i \log(p_i) \qquad (1).$$

$$\sum_{i=1}^{n} p_i = 1$$

It is a measure of uncertainty in information formulated in terms of probability theory. Entropy measure as weighting calculation method is used which weight includes the parameter that describes how much different alternatives approach one another in respect to a certain attribute. The greater the value of the entropy, the smaller the entropy weight, then the smaller the different alternatives in this specific attribute, and the less information the specific attribute provides, and the less important this attribute becomes in decision making process(in Table 3).

Table 3. Entropy based Feature Selection Algorithm

Procedure Select_Features (D, Q, a, ψ)

D : the whole set
D_r : first element in D
D_{nr} : last element in D
ψ : a predefined threshold
F : generate features(Q)

For each f ∈F
Index[f]=0,
ψ={threshold(f,d) | d∈ $D_r \cup D_{nr}$ }
For each t∈ψ
$S_{\ominus}$ ={d∈ $D_r \cup D_{nr}$ | threshold (f,d)>t}
S_{Θ} ={d∈ $D_r \cup D_{nr}$ | threshold (f,d)≤t}
$Index_t$=1-Entropy $((S_{\ominus})* \frac{|S_{\ominus}|}{|D_r \cup D_{nr}|})$-Entropy$((S_{\Theta})* \frac{|S_{\Theta}|}{|D_r \cup D_{nr}|})$
Index[f]=max(Index[f], $Index_t$)
Return

In Table 3 represents the whole set and A attribute consider the entropy method then evaluate the index value for each data. Then feature selection is realized with this algorithm.

In Table 4 realizes the query based learning process with performance index. Similarity measure, distance matrix and ranking process are used for evaluate the efficiency of query based learning process. Rule base data obtained from entropy based classification. Then results are evaluated with similarity measure.

Mean absolute error is used to measure similarity of query values; sometimes the square root is taken to give it the same dimensions as the predicted value itself. Mean absolute error averages the magnitude of the individual errors without taking account of their sign.

$$\text{Mean absolute error} = \frac{\left|p_1 - a_1\right| + \ldots + \left|p_n - a_n\right|}{n} \qquad (2)$$

Mean absolute error averages the level of the individual errors without taking account of their sign.

Especially distance matrix measures the relative of among data then assigns the rules. Distance based matrix occurs from terms which can represent each others distance value that it is determine the data relation the other say, include the next rule base.

Table 4. Query Based Learning Algorithm

Input

D : the whole set
A_1 : the features of training set
A_2 : the features of testing set
C : the target set
ψ : a predefined threshold

Output
S_{max} : the maximum subset

1. **For** i=1 to m
   Calculate $Index_{a_i,c}$ for a_i
   End
2. **For** j=k+1 to m
   For j=1 to k
   Calculate $r_{(a_i,a_k)}$ for a_i
   If $d_{(a_i,a_j)} \geq r_{a_i,c}$
   Remove a_i
   Else
   Append a_i to A_2
   Keep the biggest $Index_{a_i,a_i}$ for a_i
   End
   End
   End
3. **Sort** A_2 in ascending values of $Index_{a_i,a_i}$
4. $a_k \leftarrow$ get the fist element of A_2
5. **Do** begin
   Set $S' \leftarrow S_{max} \setminus \{a_k\}$
   Train the model with S'
   If the performance $P \leq \psi$ remove a_k from S_{max}
   $a_k \leftarrow$ get the next element of A_2 preceding a_p

   Until a_k==NULL or $P > \psi$
6. **Return** S_{max}

$$Dist(A_2,A_2) = \frac{\Xi Terms(A_1) \Delta Terms(A_2)}{(\Xi Terms(A_1) \cup Terms(A_2)) + (\Xi Terms(A_1) \cap Terms(A_2))} \quad (3)$$

In here A_1 represents the training set and A_2 represents the test set., Ξ stands for number of and Δ for the symmetrical difference between the two sets. This distance formula gives to the similarity weight value of distance elements. Consequently, for two elements that, the distance value is 1, the highest possible value, whereas for two genes sharing exactly the same set of the sets, the distance value is 0, the lowest possible value. All possible binary pairs of genes from the dataset are considered, and the distances between their sets are calculated, resulting in a distance matrix. Second, this matrix is processed with a clustering algorithm, in which the leaves correspond to input elements. The relevance of each term associated to each class is then calculated using formula 3.

The user-oriented learning system is proposed namely the Query Based Learning System (QBLS), which is based on a data centric model with extensions to provide support for user interaction. The proposed algorithm is evaluated in terms of selected features and the learning accuracy.

5 Application

In this section iris.dat set used that features used in query based learning[33]. The data report four characteristics (sepal width, sepal length, petal width and petal length) of three species of Iris flower. All measurements are lengths in cms. Feature selection based algorithm applied to iris dataset and evaluated the entropy based weight values in Table 5. Then the distance matrix applied the whole dataset and class distribution and percentage values are listed in Table6.

Table 5. Iris feature's entropy based weights

Features	Entropy Based Weights
Petal_width	0.6260
Petal_length	0.3217
Sepal_width	0.0263
Sepal_length	0.0260

Table 6. Each features class numbers and percentages

Features	Class #	Numbers and Percentage									
Petal_width	2	Class 1	56 (37,3%)	Class 2	94 (62,7%)						
	3	Class 1	51 (34,0%)	Class 2	67 (44,7%)	Class 3	32 (21,3%)				
	5	Class 1	51 (34,0%)	Class 2	20 (13,3%)	Class 3	35 (23,3%)	Class 4	16 (10,7%)	Class 5	28 (18,7%)
Petal_length	2	Class 1	63 (42,0%)	Class 2	87 (58,0%)						
	3	Class 1	51 (34,0%)	Class 2	66 (44,0%)	Class 3	33 (22,0%)				
	5	Class 1	51 (34,0%)	Class 2	8 (5,3%)	Class 3	38 (25,4%)	Class 4	5 (3,3%)	Class 5	48 (32,0%)
Sepal_width	2	Class 1	99 (66,0%)	Class 2	51 (34,0%)						
	3	Class 1	99 (66,0%)	Class 2	39 (26,0%	Class 3	12 (8,0%)				
	5	Class 1	5 (3,3%)	Class 2	5 (3,3%	Class 3	33 (22,0%)	Class 4	94 (62,7%	Class 5	13 (8,7)
Sepal_length	2	Class 1	58 (38,7%)	Class 2	92 (61,3%)						
	3	Class 1	52 (34,7%)	Class 2	74 (49,3%	Class 3	24 (16,0%)				
	5	Class 1	51 (34,0%)	Class 2	23 (15,3%	Class 3	6 (4,0%)	Class 4	27 (18,0%	Class 5	43 (28,7%)

Table 7. Each features Learning Efficiency and Query Based Learning Result

Features	Learning Efficiency					Query Based Learning Result			
	class 1	class 2	class 3	class 4	class 5	Petal_width	Petal_length	Sepal_width	Sepal_length
Petal_width	0,8460	1	0,4998	0,4242	0,3791	2,5	5,7	3,3	6,7
Petal_length	1	0,8479	0,5667	0,4829	0,3935	0,2	1	3,6	4,6
Sepal_width	1	0,8232	0,4697	0,5025	0,2951	1	3,5	2	5
Sepal_length	1	0,8848	0,5194	0,5699	0,3824	0,1	1,1	3	4,3

Table 8. Each features classification efficiency

Features	Class		Classification Efficiency								
Petal_width	2	Class 1	4,4646	Class 2	3,7260						
	3	Class 1	4,4646	Class 2	1,7550	Class 3	3,7260				
	5	Class 1	4,4646	Class 2	3,7260	Class 3	1,7550	Class 4	1,5115	Class 5	2,4124
Petal_length	2	Class 1	4,4646	Class 2	5,4784						
	3	Class 1	4,4646	Class 2	1,4410	Class 3	5,4784				
	5	Class 1	4,4646	Class 2	5,4784	Class 3	1,4410	Class 4	1,5949	Class 5	2,6372
Sepal_width	2	Class 1	1,4818	Class 2	3,4420						
	3	Class 1	1,4818	Class 2	3,9521	Class 3	3,4420				
	5	Class 1	1,4818	Class 2	3,4420	Class 3	3,9521	Class 4	1,4402	Class 5	3,2397
Sepal_length	2	Class 1	4,4362	Class 2	4,2103						
	3	Class 1	4,4362	Class 2	2,5454	Class 3	4,2103				
	5	Class 1	4,4362	Class 2	4,2103	Class 3	2,5454	Class 4	1,4427	Class 5	1,9180

Then query based algorithm (in Table 2) applied and obtained the Table 5 and 6. In Table 7 shows the learning efficiency values and query based learning results. Table 8 represents the each features classification efficiency.

6 Conclusion

This study contributes to new prospect for agent learning methodology, which is consider the query based system model and obtaining learning rules for making decision on a great extent on the use of information from objective reality with query based learning in multi agent systems with model driven approach.

Query based learning mechanism focus on the modeling of autonomous intelligent behavior, which activities, supports and/or executes the state changes. The following steps are based modeling the suggested system:

Step 1 Explore environment: The environment of the system is explored to enrich and extend the queried knowledge such that new experience is build up.
Step 2 Query knowledge: Obtained explored environment then query based algorithm used and converted to the learning rules.
Step 3. Learn results: Learning rules evaluated with decision makers experience and take new decisions for the system.

7 References

[1] Song, E., Yin, S., Ray, I., "Using UML to model relational database operations", Computer Standards & Interfaces Volume 29, Issue 3, March 2007, Pages 343-354

[2] Philippi, S., "Automatic code generation from high-level Petri-Nets for model driven systems engineering", The Jpurnal of System and Software 79(2006) 1444-1455.

[3] He, B., Chang, K.C., Han, J.,"Discovering Complex Matching across Web Query Interfaces: A Correlation Mining Approach", Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD'04, August 22-25, 2004, Seattle, Washington, USA.

[4] Jim, T., Suciu D., "Dynamically Distributed Query Evaluation", Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, Santa Barbara, California, United States, Pg: 28 – 39, 2001ISBN:1-58113-361-8

[5] Hsu, C., Knoblock, C.A., "Learning Database Abstractions for Query Reformulation", in Proceedings of the AAAI Workshop on Knowledge Discovery in Databases, Washington, D.C., 1993.

[6] Juve, G., Francoeur, H., Gordon, L., Jhatakia, S., Sharma, N., "Creating A Virtual Fault Database Using Ontologies", American Geophysical Union, Fall Meeting 2003, abstract #ED41B-1169, 12/2003.

[7] Cybenko, G., Berk, V., Crespi, V., Gray, R., Jiang, G., "An overview of Process Query Systems", Proceedings of SPIE Defense and Security Symposium, 12-16 April 2004, Orlando, Florida, USA.

[8] Deshpande, A., Guestrin, C., Hong, W., Madden, S., "Exploiting Correlated Attributes in Acquisitional Query Processing", Intel Research, IRB-TR-04-008, Jun. 2, 2004.

[9] Bernstein, P.A., Goodman, N., Wong, E., Reeve, C.L., Rothnie, J.B.,"Query Processing in a System for Distributed Databases (SDD-1)", ACM Transactions on Database Systems, Vol.6, No.4, December 1981, pg. 602-625.

[10] Genet, B., Hinze, A., "Open Issues in Semantic Query Optimization in Related DBMS", IV. Working paper series (University of Waikato. Dept. of Computer Science); 2004/10.

[11] Tatarinov, I., Ives, Z., Madhavan, J., Halevy A., Suciu, D., Dalvi, N., Dong, X.L., Kadiyska, Y., Miklau, G., Mork, P., "The Piazza Peer Data Management Project", SIGMOD Rec., 32(3):47–52, 2003.

[12] Harren, M., Hellerstein, J.M., Huebsch, R., Lao, B.T., Shenker, S., Stoica, I., "Complex Queries in DHT-based Peer-to-Peer Networks",International Workshop on Peer-to-Peer Systems (IPTPS), IEEE, Cambridge, MA, USA, mar, 2002,"http://www.cs.rice.edu/Conferences/IPTPS02

[13] Idreos, S., Tryfonopoulos, C., Koubarakis, M., Drougas, Y., "Query Processing in Super-Peer Networks with Languages Based on Information Retrieval: the P2P-DIET Approach", In Proceedings of the International Workshop on Peer-to-Peer Computing and DataBases (P2PDB 2004), Heraklion Greece, March 2004.

[14] Ben Necib, C., Freytag, J.C., "Query Processing Using Ontologies", Proceedings of the 17th Conference on Advanced Information Systems Engineering (CAISE'05), Porto, Portugal, 2005/06

[15] Turgay, S., Yaman, F., "Intelligent Query Answering Mechanism in Multi Agent Systems" chapter in "Encyclopedia of Artificial Intelligence", Editors: Juan R. Rabuñal, Julián Dorado and Alejandro Pazos, Dept. of Information & Communications Technologies, University of A Coruña, Spain, Idea Publishing group (Information Science Reference, May 2008).

[16] Hoschek, W., "Query Processing in Containers Hosting Virtual Peer-to-Peer Nodes", Int'l. Conf. on

Information Systems and Databases (ISDB 2002), Tokyo, Japan, September 2002.

[17] McClean, S., Scotney, B., Rutjes, H., Hartkamp, J., "Metadata with a MISSION: Using Metadat to Query Distributed Statistical Meta-Information Systems", 2003 Dublin Core Conference: Supporting Communities of Discourse and Practice-Metadata Research&Applications, DC-2003, 28 September-2 October 2003, Seattle, Washington USA.

[18] Turgay, S., "A Multi-agent system approach for distance learning architecture", The Turkish Online Journal of Educational Technology-TOJET October 2005 ISSN: 1303-6521 Volume 4, Issue 4, Article 3.

[19] Doherthy, P., Szalas ,A., Lukaszewicz, W. "Approximate Databases and Query Techniques for Agents with Heterogeneous Ontologies and Perceptive Capabilities", Proceedings on the 9th International Conference on Principles of Knowledge Representation and Reasoning, 2004.

[20] Doherty, P., Lukaszewicz, W., Szalas, A., "Approximate Databases and Query Techniques for Agents with Heterogeneous Perceptual Capabilities", Proceedings on the 7th International Conference on Information Fusion, 2004.

[21] Purvia, M., Cranefield, S., Bush, G., Carter, D., "The NZDIS Project: an Agent-Based Distributed Information Systems Architecture", in Proceedings of the Hawaii International Conference on System Sciences, January 4-7, 2000, Maui, Hawaii.

[22] Jiao, W., Mei, H., "Automated adaptations to dynamic software architectures by using autonomous agents", Engineering Applications of Artificial Intelligence 17(2004), 749-770.

[23] Chunlin, L., Layuan, L., "Agent framework to support the computational grid", The Journal of Systems and Software 70(2004) 177-187.

[24] Araragi, T., Attie, P., Keidar, I., Kogure, K., Luchangco, V., Lynch, N., Mano, K., "On Formal Modeling of Agent Computations", FAABS2000, LNAI 1871, pp. 48-62, 2001, Springer-Verlag, Berlin Heidelberg 2001.

[25] Kirby, J., "Model-driven Agile Development of Reactive Multi-Agent Systems", in Proceedings, 30 th Annual International Computer Software and Applications Conference (COMPSAC 2006) September 2006, Chicago, IL, USA.

[26] Pavlin, G., Oude P., Maris, M., Hood, T., "Distributed Perception Networks : An Architecture for Information Fusion Systems Based on Causal Probabilistic Models", International Conference on Multisensor Fusion and Integration for Intelligent Systems, Heidelberg, Germany, 2006.

[27] Zhu, H., Shan, L., "Caste-Centric Modelling of Multi-Agent Systems: The CAMCE Modelling Language and Automated Tools", in Beydeda, S. and Gruhn, V. (eds) Model-driven Software Development, Research and Practice in Software Engineering, Vol.II, Springer, 2005, pp57-89.

[28] Sansores, C., Pavon, J., "Agent-Based Simulation Replication: A Model Driven Architecture Approach", A. Gelbukh, A. de Albornoz and H. Terashima (Eds): MICAI 2005, LNAI 3789, pp.244-253, 2005.

[29] Cranefield, S., Pan, J., "Bridging and Gap Between the Model-Driven Architecture and Ontology Engineering", The Information Science, Discussion Paper Series, University of Otago, Department of Information Science, Number 2005/12, December 2005, ISSN 1172-6024.

[30] Grassi, V., Mirandola, R., Sabetta, A., "Filling the gap between design and performance/reliability models of component-based systems: A model driven approach", The Journal of Systems and Software xxx (2006) xxx.

[31] Amor, M., Fuentes, L., Vallecillo, A., "Bridging the Gap Between Agent-Oriented Design and Implementation Using MDA", AOSE 2004, LNCS 3382, pp.93-108, 2005, Springer-Verlag, Berlin Heidelberg 2005

[32] Xia, P., Chrysanthis, P.K., Labrinidis, A., "Similarity-Aware Query Processing in Sensor Networks", in Proc. of the 14th International Workshop on Parallel and Distributed Real-Time Systems (WPDRTS'06), April 2006, Rhodes, Greece.

[33]The data originally appeared in Fisher, R. A. (1936). "The Use of Multiple Measurements in Axonomic Problems," Annals of Eugenics 7, 179-188.

Milestone State Formulation Methods

Hyungoo Han

Department of Computer Science & Engineering
Hankuk University of Foreign Studies
89 Wangsan-ri Mohyeon Cheoin-gu Yongin-si, Gyeonggi-do 449-791 South Korea

Abstract- *An intelligent robot generates a plan to achieve a goal in a problem domain. A plan is a sequence of robot actions that accomplish a given mission by being successfully executed. However, in the real world, a robot may encounter unexpected situations and may not execute its actions. A plan repairing method is required in such situations for a robot to accomplish its given mission. Two basic procedures for handling such situations are generating a new plan and repairing an existing plan. The re-planning procedure can cause time to be lost generating a new plan while discarding an existing one. The repair procedure must allocate a large storage area to preserve every expected state transformation to adjust the unexpected changes to normal states. Plan repair with milestone states is an alternative procedure to cope with the situation. It retains the advantages of the other two procedures. This paper proposes progressive and regressive methods of formulating milestone states. A method of assigning weighting values on conditions that compose a milestone state is also proposed. The task to repair a plan employs the weighting values as its job priority. The regressive method formulates less complex milestone states and leads to the conditions of a milestone state to take pertinent weighting values for an efficient way to repair a plan.*

Keywords: planning, intelligent agents, plan repair, milestone states, weighting values, artificial.

I. INTRODUCTION

Intelligent robots, like humans, generate a plan to achieve a goal in a problem domain before they execute it [1,2,3]. A plan is a sequence of robotic actions that accomplishes a given goal by being executed successfully. The planning process is a state transformation process from an initial state to a goal state. I have modified the definition of actions of the Stanford Research Institute Problem Solver (STRIPS) slightly [4]. Each action consists of four components: a precondition formula, a set of positive post-conditions, a set of negative post-conditions, and a set of still-conditions. The precondition formula is a conjunction of prerequisite conditions for an action to be triggered. Both the set of positive and negative post-conditions are created, as the result of firing an action; the positive set consists of true or new conditions and the negative set consists of false or deleted conditions

The success of a plan is based on the consecutive and successful executions of the individual actions of a robot. However, in the real world a robot may be confronted with unexpected situations – termed error states in this paper - that may occur due to device failures or malfunctions, inconsistent sensing data, or unanticipated environmental changes [3]. Planning to consider all possible error states is almost impossible; a robot should be able to fix the error states whilst executing the plan [3,5].

The two basic approaches for handling error states are generating a new plan and repairing an existing plan. [3,6,7]. The re-planning approach discards the current plan and regenerates a plan in its current state, when an error state occurs [3,7]. This procedure builds a new plan that transforms the error state to the goal state; the plan repairing process is simple. However, this re-planning approach can lose time, since it builds a new plan, discarding the existing plan.

The second approach is regionally repairing a plan [3,8]. A robot must know the exact descriptions of an expected normal state, which would have been produced if the error state had not occurred, to regionally repair a plan when an error state is encountered. The regional repair task is the process of building a partial plan, which transforms the error state to the expected normal state, without discarding the current plan. Even though this method has the advantage of reusing the existing current plan, it must incur a large cost to store those expected normal states. A robot may keep track of every state transformation and produce each expected normal state internally, whenever it triggers an action, not to store all the expected states. However, this process will confuse a robot in a real time environment of plan execution, because the process of tracking state transformations to produce expected states is not simple; it is time consuming.

Plan repair with milestone states is another procedure [6]. This procedure has the benefits of the previous two procedures. The idea of this procedure is not to store all the expected normal states, unlike the regional plan repair procedure, but selecting and storing an appropriate number of expected normal states, as milestone states. With this procedure, a robot will choose a milestone state, which appears behind and is nearest to the current error state, and build a partial plan to transform the error state to

the chosen milestone state, discarding actions between the current state and the milestone state only.

This paper proposes progressive and regressive methods to formulate milestone states for the last procedure. A method for assigning weighting values on conditions that compose a milestone state is also proposed. The process for repairing a plan employs the weighting values as its job priority. The regressive method formulates less complex milestone states and leads the conditions of a milestone state to take more pertinent weighting values for an effective and efficient handling procedure to repair a plan.

II. RELATED WORK

The three procedures to handle unexpected environmental changes are stated in section I. This section details the plan repair with milestone states. In general, the regional plan repair procedure is more efficient for fixing an error state than is the re-planning procedure. However, theoretically the former is not always better than the latter [8,9]. The re-planning procedure wastes the existing plan and consumes more time to generate a new plan. The regional plan repair procedure also must store all the expected normal states. In contrast, the procedure with milestone states reuses the existing plan, as much as possible, and stores as few of the expected normal states as possible.

Fig. 1 depicts the expected normal states and selected milestone states of a plan. The circles, regardless of their size, denote the expected normal states. Executing the plan transforms the initial state S_1 to the goal state S_n in a given problem domain, when all the actions of the plan are triggered consecutively and successfully. Action a_i is triggered in state S_i and produces S_{i+1}. That is, if a robot

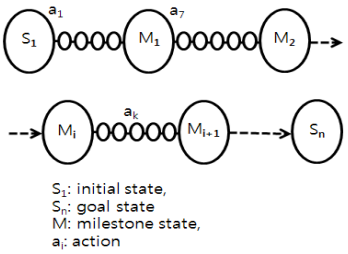


Fig. 1. A plan space and milestone states

recognizes that every precondition of a_i is sound as expected in S_i, it triggers a_i in S_i with transforming S_i to S_{i+1}. In Fig. 1, every 6th state is selected as a milestone state for plan repair. Big circles with M including the goal state S_n in the figure are milestone states and will be stored in the robot. Suppose that a robot finds that it cannot trigger action a_k in the current state S_k, i. e. the S_k is an error state, it will choose the first following milestone state M_{i+1} and generate a partial plan that will transform the error state S_k to M_{i+1}. Under this situation, with the re-planning procedure the robot will throw away the entire plan and generate a new plan that transforms S_k to the goal state S_n. Note that when a_k is in the forepart of the plan the robot has to consume much effort to generate a new lengthy plan. The robot should know the perfect descriptions of the expected normal state with the regional plan repair procedure, as mentioned in the previous section.

In Fig. 1, every 6th state is selected as a milestone state. Determination of the appropriate number of milestone states to repair error states is not within the scope of this paper. Note that when no milestone state is selected, the repair procedure with milestone states becomes the re-planning procedure and all the states are selected as milestone states, the procedure becomes the regional plan repair procedure. If a robot selects more milestone states it must allocate more storage and, if it selects less milestone states, it has to discard more actions and spend more time to generate a partial plan. Therefore, it is not a simple task to determine how many milestone states should be selected to repair error states; this may depend on different factors in problem domains.

III. TWO METHODS TO FORMULATE MILESTONE STATES

III-1. Problem domain and a blocks world

I employed the blocks world in Fig. 2 as a problem domain to describe how the two methods work to formulate milestone states. It is also used to compare the efficiency of their roles in repairing error states and explain the mechanism of assigning weighting values on conditions in the problem domain. In the figure, the plan with 28 actions is generated from the given initial and goal state descriptions. Box B and box C are depicted with dotted lines, because the conditions related to the two

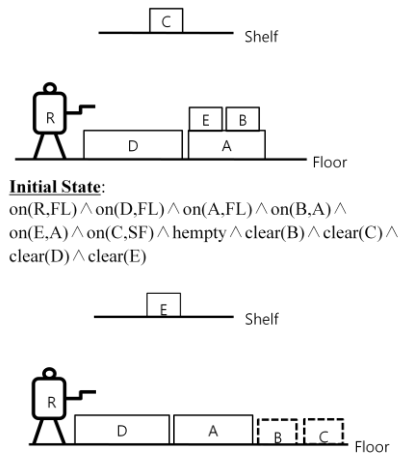


Fig. 2. Blocks world problem domain

boxes are irrelevant to the goal state and the goal state description in the figure does not contain any conditions relevant to the two boxes.

Some constraints are imposed in the problem domain. Two different actions, Pick and Unstack, are used to hold a box in the hand of the robot. Unstack(X,Y) is to remove X from Y when Y is a box and both robot and Y are on the same object. Conversely, Pick(X,Y) picks up X from Y when both the robot and X are on Y. Stack(X,Y) and Put(X,Y) are the actions to place X on Y. Stack(X,Y) stacks X on Y when the robot and Y are on the same object. Put(X,Y) is to place X on Y when the robot is on Y. The robot must empty its hand to hold a box, climb, or come down a box. Only one box can be placed on the shelf; to reach a box on the shelf, the robot must stack box A on box D and get on the pile of the two boxes. The robot cannot reach a box on a pile of multiple boxes. The robot, box A, and another box, or the robot and two boxes other than box A, can be placed on box D simultaneously. The robot and a box, or two boxes other than box D, can simultaneously be on box A. There is always room on the floor for the robot and boxes.

III-2. State space modeling of a plan

A conceptual model for state transition T is defined as a 4-tuple system, since the execution of a plan is concerned with firing actions to change the states of a problem domain. $T = (S, A, C, \tau)$[10]. S is a finite set of domain states, A is a finite set of robot actions, C is a finite set of conditions that comprise the domain states, and $\tau: S \times A \times C \rightarrow S$ is a state transition function. τ is represented as $\tau(s_i, a_i, \text{precond}(a_i)) \rightarrow s_{i+1}$. When the preconditions of an action a_i, which are denoted as $\text{precond}(a_i)$ are satisfied in the state s_i, τ triggers a_i in s_i, and causes the state transition from s_i to s_{i+1}. This new state s_{i+1} consists of $\text{postcond}(a_i)$, $\text{postcond}^-(a_i)$, and $\text{stillcond}(a_i)$.

$\text{Postcond}(a_i)$ has conditions that are satisfied in compliance with action a_i. $\text{Postcond}^-(a_i)$ has conditions that may be deleted or negated with action a_i. $\text{Stillcond}(a_i)$ has conditions that are members of state s_i, and irrelevant to the execution of the action a_i. This stillcond of an action has the conditions created by some previously executed actions and may have preconditions of following actions. The length of a plan is the number of the actions that compose a plan. The number of produced states exceeds that of the plan length by one, when the execution of all actions of the plan is carried out successfully. Both the progressive and the regressive methods produce the same number of states.

A milestone state is one of the domain states produced by applying function τ to the actions of a plan. An appropriate number of domain states are selected as milestone states. Determination of an appropriate number may be a distinct research topic of the plan repair procedure with milestone states. Therefore, the methods to build a pool of domain states, from which the milestone states will be selected, are proposed in this paper.

III-3. Progressive method

The progressive method transforms domain states forward consecutively. A progressive transition function τ^f to produce states forward is defined as $\tau^f : S \times A \times C \rightarrow F$. F is a finite set of domain states produced by applying τ^f to actions of A and is a sub set of S. A progressive transition function τ^f is defined as:

$$\tau^f(f_i, a_i, \text{precond}(a_i)) \rightarrow f_{i+1}, \text{ where}$$

$$f_{i+1} = (\text{postcond}(a_i) \wedge \text{postcond}^-(a_i)) \vee \text{stillcond}(a_i),$$

$$f_{i+1} = f_{i+1} - (\forall c_i \in f_{i+1}, \ni (c_i \in f_{i+1}) \wedge (\neg c_i \in f_{i+1})).$$

Note that f_i and f_{i+1} are the elements of F, f_1 is an initial state, and f_{n+1} is a goal state that is produced with $\tau^f(f_n, a_n, \text{precond}(a_n))$ when the length of a plan is n. The last expression states that a condition and its negation cannot reside in a state simultaneously.

The stillcond of an action consists of both the irrelevant conditions to the action and the conditions unaffected by the execution of the following actions, up to the next milestone state. In some cases, a stillcond of an action may have conditions that remain unchanged by the following actions up to the goal state. These unaffected conditions naturally exist in the milestone states that are formulated in compliance with the progressive method. The unaffected conditions will appear more in the milestone states as the plan execution proceeds, since it is possible that some of the postcond or postcond^- of an action can become the stillcond of following actions.

Furthermore, these phenomena will be amplified, when a given problem domain is complex and the domain states become more complex. Some unaffected conditions may not be preconditions of the following actions or unnecessary conditions for a robot to accomplish its mission. Meanwhile, the plan repair task must generate a complete partial plan that satisfies all the conditions of the milestone state chosen to fix an error state, regardless of their necessity. Therefore, a robot may make additional effort to generate and execute a partial plan, with the milestone states formulated by the progressive method. It will be hard to expect an efficient way to repair a plan.

Table 1 shows example states produced by the two methods. Both methods will produce 29 states with the example blocks world domain in section III-1.

III-4. Regressive method

The regressive method transforms domain states backward consecutively. A regressive transition function τ^b to produce states backward is defined as $\tau^b: S \times A \times C \rightarrow B$. B is a finite set of domain states produced

byapplying τ^b to actions of A and is a subset of S. When the plan length is n, b_1 and b_{n+1} are elements of B and the initial state and the final state, respectively, as for the progressive transition function in the previous section.

TABLE 1. Example states

	States produced by progressive method		States produced by regressive method
f_1	on(R,FL) ∧ hempty ∧ on(A,FL) ∧ on(B,A) ∧ clear(B) ∧ on(C,SF) ∧ clear(C) ∧ on(D,FL) ∧ clear(D) ∧ on(E,A) ∧ clear(E)	b_1	on(R,FL) ∧ hempty ∧ on(A,FL) ∧ on(B,A) ∧ clear(B) ∧ on(C,SF) ∧ clear(C) ∧ on(D,FL) ∧ clear(D) ∧ on(E,A) ∧ clear(E)
f_2	on(R,FL) ∧ on(A,FL) ∧ clear(A) ∧ hold(B) ∧ on(C,SF) ∧ clear(C) ∧ on(D,FL) ∧ clear(D) ∧ on(E,A) ∧ clear(E)	b_2	on(R,FL) ∧ on(A,FL) ∧ clear(A) ∧ hold(B) ∧ on(C,SF) ∧ clear(C) ∧ on(D,FL) ∧ clear(D) ∧ on(E,A) ∧ clear(E)
..	.....	..	.....
f_7	on(R,FL) ∧ hempty ∧ on(A,D) ∧ clear(A) ∧ on(B,FL) ∧ clear(B) ∧ on(C,SF) ∧ clear(C) ∧ on(D,FL) ∧ clear(D) ∧ on(E,FL) ∧ clear(E)	b_7	on(R,FL) ∧ hempty ∧ on(A,D) ∧ clear(A) ∧ on(C,SF) ∧ clear(C) ∧ on(D,FL) ∧ clear(D) ∧ on(E,FL) ∧ clear(E)
f_8	on(R,D) ∧ hempty ∧ on(A,D) ∧ clear(A) ∧ on(B,FL) ∧ clear(B) ∧ on(C,SF) ∧ clear(C) ∧ on(D,FL) ∧ clear(D) ∧ on(E,FL) ∧ clear(E)	b_8	on(R,D) ∧ hempty ∧ on(A,D) ∧ clear(A) ∧ on(C,SF) ∧ clear(C) ∧ on(D,FL) ∧ clear(D) ∧ on(E,FL) ∧ clear(E)
..	.....	..	.....
f_{28}	on(R,FL) ∧ hold(A) ∧ on(B,FL) ∧ clear(B) ∧ on(C,FL) ∧ clear(C) ∧ on(D,FL) ∧ clear(D) ∧ on(E,SF) ∧ clear(E)	b_{28}	on(R,FL) ∧ hold(A) ∧ on(D,FL) ∧ clear(D) ∧ on(E,SF) ∧ clear(E)
f_{29}	on(R,FL) ∧ hempty ∧ on(A,FL) ∧ clear(A) ∧ on(B,FL) ∧ clear(B) ∧ on(C,FL) ∧ clear(C) ∧ on(D,FL) ∧ clear(D) ∧ on(E,SF) ∧ clear(E)	b_{29}	on(R,FL) ∧ hempty ∧ on(A,FL) ∧ clear(A) ∧ on(D,FL) ∧ clear(D) ∧ on(E,SF) ∧ clear(E)

A regressive transition function τ^b is defined as:

$\tau^b(b_i, a_{i-1}, \text{precond}(a_{i-1})) \rightarrow b_{i-1}$, where

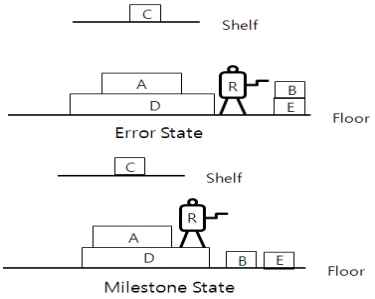
$b_{i-1} = \text{precond}(a_{i-1}) \wedge (\forall c_i \in b_i, c_i \notin (\text{postcond}(a_{i-1}) \vee \text{postcond}^-(a_{i-1})))$,

$b_{i-1} = b_{i-1} - (\forall c_i \in b_{i-1}, \ni (c_i \in b_{i-1}) \wedge (\neg c_i \in b_{i-1}))$.

b_{i-1} must have $\text{precond}(a_{i-1})$, which is the set of prerequisite conditions of ai_{i-1}, since action a_{i-1} is to be triggered in state b_{i-1} to produce state b_i. State b_{i-1} must not have the $\text{postcond}(a_{i-1})$ or the $\text{postcond}^-(a_{i-1})$, which are in state b_i, because the execution of a_{i-1} will create these conditions and produce state b_i. The last expression states that a condition and its negation cannot reside in a state simultaneously. The stillconds of actions will not have unnecessary conditions, unlike the progressive method, but have the conditions that compose the goal state or preconditions of following actions instead, since the regressive function produces states backwards. Therefore, the unnecessary conditions may also disappear in the milestone states selected from those states produced by the regressive method. The plan repair task with this method will be much lighter and simpler than the task with the progressive method. The lengths of the partial plans to repair the error state in Fig. 3 reveal the efficiency of the regressive method.

III-5. An example for repairing an error state

Fig. 3 shows an example of an error state and a selected milestone state. The robot is about to trigger the 8th action, Geton(D,A), of the plan in Fig. 2 and has encountered an error state, as in Fig. 3. Since the precondition, on(R, D), of the action is not true in the error state, the robot cannot trigger the 8th action. The robot will choose the milestone state in Fig. 3 and generates a partial plan that will transform the error state to the milestone state. Two different forms, Mf_2 by the progressive method and Mb_2 by the regressive method, are described below the picture of the milestone state, from Table 3 in section IV-1. The partial plan for Mf_2 has more actions to repair the error state than the partial plan for Mb_2



Milestone state($\mathbf{M_{f2}}$) by progressive method:
on(R,D)∧hempty∧on(A,D)∧clear(A)∧on(B,FL)∧clear(B)∧on(C,SF)∧clear(C)∧on(D,FL)∧clear(D)∧on(E,FL)∧clear(E)
Partial plan for repairing error state with **Mf_2**
Unstack(B,E), Put(B,FL), Geton(FL,D)

Milestone state($\mathbf{M_{b2}}$) by regressive method:
on(R,D)∧hempty∧on(A,D)∧clear(A)∧on(C,SF)∧clear(C)∧on(D,FL)∧clear(D)∧on(E,FL)∧clear(**E**)
Partial plan for repairing error state with **Mb_2**
Geton(FL,D)

Fig. 3. An error state and plan repair

III-6. Weighting values of milestone state conditions

A weighting value is assigned to each condition composing a milestone state to provide a guide to fix an error state with the milestone states. The task will be able to fix error states efficiently, since the order of fixing the conditions for the plan repair task can be determined by the weighting values. The principle of assigning weighting values on conditions is that for a condition appearing in two consecutive milestone states simultaneously, the condition appearing in the latter milestone state gets additional weighting values. Note that an initial state is employed as a milestone state to pair up to the first milestone state.

The conditions appearing in two consecutive milestone states simultaneously may not be the preconditions of the actions between the two milestone states, but may be the preconditions of the following actions of the latter milestone state or the conditions that compose the goal state. These conditions are usually created by the actions that are executed before the first milestone state of the two consecutive milestone states. The continuously appearing conditions may be preserved during the execution of the actions between the two consecutive milestone states and have higher weighting values than other conditions of the rear milestone state, since the unnecessary conditions are removed from milestone states formulated by the

regressive method.

The following expressions assign weighting values on conditions when it is assumed that W is a weighting value of a condition, RC is a condition related to a robot, and c is a condition of a milestone state M.

$\forall c_i \in M_i, (c_i \in M_{i+1} \wedge c_i \notin RC) \rightarrow W(c_i \in M_{i+1}) + 2$

$\forall c_i \in M_i, (c_i \in M_{i+1} \wedge c_i \in RC) \rightarrow W(c_i \in M_{i+1}) + 1$

$\forall c_i \in M_{i+1}, c_i \notin M_i \rightarrow W(c_i \in M_{i+1}) = 0$

RC is increased by 1 not 2, as a robot is the main agent handling the domain conditions and it has to handle the conditions that are not related to it, prior to handling the RC. When a new condition appears in the rear milestone state, a zero weighting value is given to it.

IV. COMPARISON OF THE TWO METHODS

IV-1. Complexity comparisons

In Table 2, the length of the plan is the number of actions of the plan in Fig. 2, the number of states is the number of states produced by the simulated execution of the plan, and the number of conditions is the total number of conditions of 29 states. Note that the 29 states will be produced by the execution of the 28 actions in the plan. The number of conditions and the average number of state conditions in the table show that the regressive method produces states that are less complex than are those of the progressive method. That is, fewer conditions are to be repaired when a plan repair task is needed.

TABLE 2. Complexity of states

Methods	Length of plan	Number of states	Number of conditions	Average
Progressive method	28	29	338	12
Regressive method	28	29	260	9

Table 3 shows the two types of milestone states selected from the 29 states that are produced by progressive and regressive transit functions, with the plan in Fig. 2.

In this paper, every 4th state, including the goal state, is chosen as a milestone state from the 29 states. The goal state is the last milestone state; there are 8 milestone states. In Table 3, Mf_i and Mb_i are the names of the milestone states formulated by the progressive and the regressive methods respectively, where i is an index of a milestone state and an integer between 1 and 8. There must be a method to pinpoint a milestone state, when an error state is encountered. The integer i becomes the index of the chosen milestone, when it satisfies the following expression, where k is the index of an action encountering an error state and p is a constant used to select milestone states.

$\min(k < i \times p + 1)$

From the problem domain, if action 6 has encountered an error state, i will be 2 by the expression $\min(6 < i \times 4 + 1)$, and the second milestone state is chosen for plan repair with milestone states.

TABLE 3. Milestone states

	Progressive method		Regressive method
Mf_1	on(R,FL)∧on(A,FL)∧clear(A)∧on(B,FL)∧clear(B)∧on(C,SF)∧clear(C)∧on(D,F),∧clear(D)∧hold(E)	Mb_1	on(R,FL)∧on(A,FL)∧clear(A)∧on(C,SF)∧clear(C)∧on(D,FL)∧clear(D)∧hold(E)
Mf_2	on(R,D)∧hempty∧on(A,D)∧clear(A)∧on(B,FL)∧clear(B)∧on(C,SF)∧clear(C)∧on(D,FL)∧clear(D)∧on(E,F)∧clear(E)	Mb_2	on(R,D)∧hempty∧on(A,D)∧clear(A)∧on(C,SF)∧clear(C)∧on(D,FL)∧clear(D)∧on(E,FL)∧clear(E)
Mf_3	on(R,D)∧hempty∧on(A,D)∧clear(A)∧on(B,FL)∧clear(B)∧on(C,A)∧clear(C)∧on(D,FL)∧clear(D)∧on(E,FL)∧clear(E)∧clear(SF)	Mb_3	on(R,D)∧hempty∧on(A,D)∧clear(A)∧on(C,A)∧clear(C)∧on(D,FL)∧clear(D)∧on(E,FL)∧clear(E)∧clear(SF)
Mf_4	on(R,FL)∧on(A,D)∧clear(A)∧on(B,FL)∧clear(B)∧hold(C)∧on(D,FL)∧clear(D)∧on(E,FL)∧clear(E)∧clear(SF)	Mb_4	on(R,FL)∧on(A,D)∧clear(A)∧hold(C)∧on(D,FL)∧clear(D)∧on(E,FL)∧clear(E)∧clear(SF)
Mf_5	on(R,D)∧hempty∧on(A,D)∧clear(A)∧on(B,FL)∧clear(B)∧on(C,FL)∧clear(C)∧on(D,FL)∧on(E,D)∧clear(E)∧clear(SF)	Mb_5	on(R,D)∧hempty∧on(A,D)∧clear(A)∧on(D,FL)∧clear(D)∧on(E,D)∧clear(E)∧clear(SF)
Mf_6	on(R,A)∧on(A,D)∧clear(A)∧on(B,FL)∧clear(B)∧on(C,FL)∧clear(C)∧on(D,FL)∧clear(D)∧hold(E)∧clear(SF)	Mb_6	on(R,A)∧on(A,D)∧clear(A)∧on(D,FL)∧clear(D)∧hold(E)∧clear(SF)
Mf_7	on(R,FL)∧hold(A)∧on(B,FL)∧clear(B)∧on(C,FL)∧clear(C)∧on(D,FL)∧clear(D)∧on(E,SF)∧clear(E)	Mb_7	on(R,FL)∧hold(A)∧on(D,FL)∧clear(D)∧on(E,SF)∧clear(E)
Mf_8	on(R,FL)∧hempty∧on(A,FL)∧clear(A)∧on(B,FL)∧clear(B)∧on(C,FL)∧clear(C)∧on(D,FL)∧clear(D)∧on(E,SF)∧clear(E)	Mb_8	on(R,FL)∧hempty∧on(A,FL)∧clear(A)∧on(D,FL)∧clear(D)∧on(E,SF)∧clear(E)

IV-2. Weighting value comparisons

The weighting values of the milestone states formulated by the progressive and the regressive methods are assigned in the same way as in section III-6. Table 4 shows the weighting values assigned to the conditions of the milestone states in Table 3. The conditions of states are the domain conditions that appear in all 29 states. In this table, each milestone state is composed only by the conditions with weighting values 0, 1, or 2. The highlighted part of the table shows that in the progressive method, high weighting values are assigned to the unnecessary conditions related to box B and box C. In contrast, in the regressive method, those conditions have low weighting values or do not appear in the milestone states. This means that the unnecessary conditions disappear in the list of conditions to fix an error state. The error state repair task will try to compose a partial plan for the conditions of a chosen milestone state in the order of their weighting values. Therefore, the regressive method will guide the error state repair task to be more efficient than will the

progressive method.

TABLE 4. Weighting values of conditions

Conditions	Milestone states															
	Progressive method								Regressive method							
	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
on(R,FL)	1			0			0	1	1			0			0	1
on(R,A)						0								0		
on(R,D)		0	2		0					0	2		0			
hempty		0	2		0			0		0	2		0			0
hold(A)							0								0	
hold(B)																
hold(C)				0								0				
hold(E)	0					0			0					0		
on(A,FL)	2							0	2							0
on(A,D)		0	2	4	6	8				0	2	4	6	8		
clear(A)	0	2	4	6	8	10		0	0	2	4	6	8	10		0
on(B,FL)	0	2	4	6	8	10	12	14								
on(B,A)																
clear(B)	2	4	6	8	10	12	14	16								
on(C,FL)					0	2	4	6								
on(C,A)			0								0					
on(C,D)																
on(C,SF)	2	4							2	4						
clear(C)	2	4	6		0	2	4	6	2	4	6					
on(D,FL)	2	4	6	8	10	12	14	16	2	4	6	8	10	12	14	16
clear(D)	2	4	6	8		0	2	4	2	4	6	8	10	12	14	16
on(E,FL)		0	2	4						0	2	4				
on(E,A)																
on(E,D)					0								0			
on(E,SF)							0	2							0	2
clear(E)		0	2	4	6		0	2		0	2	4	6		0	2
clear(SF)			0	2	4	6					0	2	4	6		

V. CONCLUSION

This paper proposes the progressive and regressive methods to formulate milestone states and the method to assign weighting values on the conditions that compose a milestone state. The progressive method forces a milestone state to have unnecessary conditions to repair error states, while the regressive method does not. Therefore, the regressive method usually formulates smaller and less complex milestone states to fix the error states. When a repair list of conditions becomes longer, the length of a new partial plan to repair the conditions on the list also becomes larger. A robot has to spend more time to generate a larger partial plan and to execute the larger partial plan. Unlike the progressive method, the regressive method causes the unnecessary conditions to assume low weighting values or disappear from the milestone states. The task to fix error states employs the weighting values as its job priority. Therefore, the regressive method to formulate milestone states provides an efficient way to repair plans. The proposed method is unsuited to path selection plans. A study on topics, such as the similarity between milestone states, the degree of proximity of locations and domain states, the heuristic approaches to repair error states, and so forth, will deserve a place in future plan repairing methods with milestone states.

REFERENCES

[1] Yongtae Do et al., 'Artificial Intelligence Concepts and Applications', 3rd Edition, SciTech, Seoul Korea, 2009

[2] Malik Ghallab, Dana Nau, Paolo Traverso. 'Automated Planning Theory and Practice', Morgan Kaufmann Publishers, New York, 2004

[3] Hyungoo Han, Kai Chang, William Day, 'A Comparison of Failure Handling Approaches for Planning Systems – Replanning vs. Recovery', Journal of Applied Intelligence, Vol. 3, Kluwer academic publishers, 275-300, 1993

[4] Richard E. Fikes, Nils J. Nilsson, 'STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving', Artificial Intelligence (2), pp.189-208, 1971

[5] Jianming Guo, Liang Liu, 'A Study of Improvement of D* Algorithms for Mobile Robot Path Planning in Partial Unknown Environments', Kybernetes Volume 39, Emerald Group Publishing Limited, Issue 6, pp935-945, 2010

[6] Hyungoo Han, 'Plan repair with Milestone States', Institute of Information Industrial Engineering, Hankuk University of Foreign Studies, Vol. 13, pp. 205-216, 2009

[7] Bernhard Nebel, Jana Koehler, 'Plan Reuse versus Plan Generation: A Theoretical and Empirical Analysis,' Artificial Intelligence, 76, pp. 427-454, 1995

[8] Roman van der Krogt, Mathijs de Weerdt, 'Plan Repair as an Extension of Planning', ICAPS, pp. 161-170, 2005

[9] C. A. Broverman and W. B. Croft, 'Reasoning about Exceptions during Plan Execution Monitoring', Proc. Natl. Conf. Artificial Intelligence, Seattle, WA, pp .190-195, 1987

[10] T. Dean and M. Wellman, 'Planning and Control', Morgan Kaufmann, 1991

SESSION

MEDICAL + HEALTH INFORMATICS AND RELATED ISSUES

Chair(s)

TBA

Prediction of Chronic Fatigue Syndrome Using Decision Tree-Based Ensemble Methods

Christine M. Bronikowski[a], Angela Weng[b], Jacob D. Furst[c], Daniela S. Raicu[d]

[a]Vanderbilt University, [b]Northwestern University, [c,d]DePaul University

1. ABSTRACT

A decision tree ensemble classifier is a machine learning technique in which several decision trees are joined in an effort to improve the classification accuracy of a prediction model. In this paper, a three-level decision tree ensemble is proposed in the classification of chronic fatigue syndrome (CFS). While existing CFS classification models are based largely on biomarker data, immunological data, and other biologically-based datasets, we are proposing to create a classification model based on patient responses to a medical survey. We found that decision tree ensemble approaches generally performed poorly in comparison to single decision tree methods. This poor performance may be explained by an accumulation of error through the levels of the ensemble, a variable-sample imbalance, the large amount of missing values present in the dataset, the use of a subjective method of data retrieval (self-report), or any combination of these factors.

Keywords: decision tree ensemble, chronic fatigue syndrome, computer-aided medical diagnosis, classification, machine learning

2. BACKGROUND AND MOTIVATION

Chronic fatigue syndrome (CFS) is a disorder characterized by at least six months of debilitating fatigue [1], in conjunction with four or more of the following symptoms: impaired memory or concentration, unrefreshing sleep, headaches, musculoskeletal pain, sore throat, and lymph node pain or tenderness [2]. The primary issue in CFS research remains whether the disease constitutes a pathologically distinct entity, or is merely a subset of disabling, but nonspecific, symptoms shared by other diseases [1].

Due to the complex nature of chronic fatigue syndrome, there exists a great need for a systematic and comprehensive method for evaluating and classifying those afflicted with the illness [1]. Because of its unknown and intricate etiology, CFS has sparked much interest in the data mining and machine learning communities, as methods in these areas hold much promise in the venue of computer-aided diagnosis. The use of machine learning algorithms allows researchers to create robust predictive models of diseases of interest. These models can help medical professionals better understand the disease and in turn, more effectively diagnose and treat those ridden with the illness [3]. In this paper, we examine how well decision-tree based ensemble methods classify chronic fatigue syndrome based strictly on responses to a medical questionnaire.

2.1 Existing machine learning techniques in the classification of chronic fatigue syndrome

There has already been much work done in incorporating machine learning techniques in the classification of CFS. Previous works employed non-linear classification schemes, such as artificial neural networks (ANNs), to tackle the CFS classification problem [4]. Artificial neural networks are computer based models that can evaluate complex relationships, such as those found among the symptoms of chronic fatigue syndrome [5]. Hanson et al. used an ANN classifier to detect immunological differences between CFS groups and non-CFS control groups. Linear classification schemes utilized prior to this work were unable to detect the differences between the two groups. However, the ANN was able to successfully identify these immunological factors. [4]. Linder et al. examined the utility of an ANN and other conventional methods in generating classification criteria for CFS and other illnesses that feature prominent fatigue, namely systematic lupus erythematosus (SLE) and fibromyalgia syndrome (FMA). The classification criteria produced by the ANN for CFS, LE, and FMA achieved the greatest accuracy [5]. The success of these ANN models suggests that non-linear classification methods are optimal for complex medical classification problems.

While it is true that artificial neural networks can evaluate complex relationships [5], the use of ANNs as a classification tool has some disadvantages. The internal function and structure of many classification tools can, for the most part, be observed and validated. Such is not the case for an artificial neural network, which employs complex relationships present in a dataset to create a "black-box" [6] system, whose internal functions are principally unknown and inexplicable. This black-box property of an ANN raises many questions about its usefulness as a diagnostic tool [6].

There have been attempts in comparing the methods used to classify chronic fatigue syndrome. Huang et al. used genetic factors known as single nucleotide polymorphisms (SNPs) to predict chronic fatigue syndrome. Three different methods of classification were applied to the SNP dataset: naive Bayes, support vector machine, and C4.5 decision tree algorithm, in conjunction with two methods of feature selection. The naive Bayes model with both types of feature selection had the best performance, but the statistical significance of these results was not assessed. Although the accuracy, specificity, and sensitivity of the predictive models were poor, this work demonstrates that machine learning algorithms hold much promise in the problem of classifying chronic fatigue syndrome [7].

Most previous work that incorporated machine learning with CFS diagnosis dealt largely with classification of the disease based on genetic data, immunological data, and other largely biologically based datasets. Very little has been done in classifying patients based on health survey data. Goertzel et al. sought to explore the relationship between chronic fatigue syndrome and allostatic load (AL) [8], a measure of the wear and tear on a body resulting from chronic stress [9]. Laboratory and clinical data were used to calculate the AL for all study participants. The AL was then compared to an individual's overall physical and mental functioning and impairment, as measured by a series of short health surveys. Researchers found that a high AL was associated with lower scores of physical and mental functioning. While this work combined both biological data and survey data, our work seeks to explore the predictive power of a classification model that uses only health survey data [8].

2.2 Decision trees in other medical classification problems

Decision trees have been used in modeling other diseases with multi-factorial and complex pathologies. Niu et al. utilized a decision tree model to identify distinct serum protein biomarkers in individuals with rheumatoid arthritis [10]. Decision trees are non-linear predictive tools that represent classification models in a hierarchical and sequential structure. An individual's inclusion in a particular class is dependent upon their data, i.e., biomarker information; tree parameters; and the rules the tree uses to classify patients [3]. Tree parameters are user-inputted and determine the depth and size of the tree; tree rules are dependent upon the algorithm selected [11]. The decision tree model for protein biomarkers achieved 85.71% sensitivity and 87.76% specificity [10].

Lofaro et al. utilized a decision tree classification model to predict chronic allograft nephropathy, a disease of transplanted tissue, from routine blood and urine tests. Decision tree models were able to achieve sensitivities of 62.5% and 81.3% and false-positive rates of 7.2% and 25%, respectively. [3]. Greco et al. applied a decision tree classifier to a population of transplant patients. The interactions between BMI and other risk factors were examined to create predictive models of graft failure. The decision tree model achieved a sensitivity of 88.2% and a specificity of 73.8% [12].

Ting et al. employed a decision tree model in the classification of acute appendicitis. Their model was built to supplement the Alvarado Scoring System (ASS), a set of criteria that assigns particular numerical values with the presence and severity of specific symptoms to determine whether an individual is a candidate for a laparotomy. While the ASS is often used by surgeons in areas of the world with limited diagnostic and imaging technology, it has often been criticized for its clinical inaccuracy. In an attempt to improve prediction accuracy, researchers took ASS data and implemented it into a top-down decision tree algorithm. Compared to an assessment of a patient's suitability for surgery with ASS data alone, the decision tree-based model exhibited overall greater accuracy, achieving a sensitivity and specificity of 94.5% and 80.5%, respectively [13].

Similarly, Farion et al. applied a decision tree algorithm to a dataset created from the medical histories of patients in order to predict the severity of their asthma symptoms. In an attempt to normalize data values and decrease noise, a fair amount of the original data was preprocessed before application of the algorithm. The decision tree model achieved a sensitivity and specificity of 84% and 71%, respectively [14]. While only preliminary, these results demonstrate the utility of a decision tree model in predicting diseases with complex pathologies and non-traditional datasets.

2.3 Ensemble learning

The literature discussed up until this point has primarily focused on classification methods that employ a single model at any given time. Ensemble classifiers comprise an exciting new area of research in machine learning and data mining [15]. Rather than the use of a single classifier (one decision tree, one artificial neural network, etc.), an ensemble classifier uses a combination of classifiers, (multiple decision trees, multiple artificial neural networks, etc.). Accuracy is especially important in classification models with medical applications, as the use of ensemble classifiers can lead to improved generalization [16]. A high percentage of false negatives increases the risk of individuals not receiving the medical care needed to treat their illness. A high percentage of false positives causes unnecessary worry and can overwhelm medical resources [17].

Ensembles of artificial neural networks have been proposed in medical classification problems [18], [19], [20], [21]. But because an ANN ensemble is composed of multiple ANNs, its comprehensibility as a machine learning technique is even worse than that of a single ANN. Zhou et al. utilized an artificial neural network ensemble to identify lung cancer cells from needle biopsies. The ANN ensemble was built with a two-level structure. The first-level ensemble was used to determine whether a cell is judged *normal* or *cancer*. The second-level ensemble was then used to judge cells labeled cancerous by the first level, where each individual network outputs *adenocarcinoma, squamous cell carcinoma, small cell carcinoma, large cell carcinoma*, or *normal*. This double-level ANN ensemble was able to achieve greater accuracy than a single ANN [18]. However, the inability to check and explain the diagnostic process of an ANN ensemble hinders the technique's acceptance in diagnostic medicine [19].

Decision trees are computationally efficient classification models [22] whose internal structure and function *can* be easily examined and validated. For this reason, they may prove to be valuable tools in complex medical classification problems. Decision tree based techniques are able to cope with a variable-sample imbalance, that is, a dataset in which the number of variables is much larger than the number of samples. Geurts et al. employed four decision tree-based ensemble methods in the classification of rheumatoid arthritis (RA) and inflammatory bowel disease (IBD). The biomarker dataset used in this paper had high-dimensional input spaces (several thousand variables), but only a small number of

samples (a few hundred). A decision tree-based classification method was thus appropriate. Ensemble classifiers generally had greater accuracies than the single-tree classifier for both RA and IBD, highlighting the flexibility and generalizability of decision tree-based ensemble classifier approaches in medical diagnosis [22].

Ozcift applied the Random Forests decision tree ensemble classifier to improve prediction of cardiac arrhythmias. The author cites success of the classifier with particular regard to its superiority in classification accuracy (90.0%) over a single tree model and its efficiency in coping with multiple classes with small sample sizes. [23] Schietgat et al. applied a similar decision tree ensemble classifier to determine the biological functions of particular segments of genes and came to a similar conclusion – the ensemble classifier had consistently higher predictive accuracy than a single tree [24]. Such findings support the application of a similar decision tree ensemble learner to the problem of CFS classification.

3. METHODS AND TECHNICAL SOLUTIONS

3.1 Materials

The dataset includes responses to a medical questionnaire administered to participants of a community-based study of chronic fatigue syndrome. There were a total of 108 individuals classified by a team of physicians into four distinct categories: CFS (N=24), ICF (N=6), Exclusion (N=50), and Control (N=28). Participants labeled 'CFS' met all requirements for the disease, as specified by the Fukuda et al. definition of chronic fatigue syndrome. Those labeled 'ICF' had clinically evaluated, unexplained chronic fatigue that did not meet criteria for CFS. Those labeled 'Exclusion' had chronic fatigue that was explained by a pre-existing medical or psychiatric condition. Individuals who did not suffer from chronic fatigue comprised the control group [1].

3.2 Pre-processing steps

The researchers who administered and collected the CFS survey data came from the Department of Psychology at DePaul University. Before the application of any machine learning techniques, several modifications needed to be made to the dataset so as to reduce noise. Researchers who collected the survey data inputted a value of '999' for all missing responses. Upon statistical analysis of the data, we found that the '999' entries were creating unusually high variances for all of the variables. Deleting the '999' entry and leaving a blank in the response column corrected this distortion. A total of 7615 missing data entries were affected.

Several items in the questionnaire were structured in the following way: "Where would you rate (symptom) on a 100-point scale?" While other items in the survey were formatted so that a respondent had between two and ten options to choose from, these 100-point scale questions allow for essentially one hundred distinct responses. Implementing percentile binning, such that for every response, "x", a new value is assigned according to the following criteria:

- 0 = symptom not present
- 1 = $0 < x \leq$ 25th percentile of the original response distribution
- 2 = $25 < x \leq$ 50th percentile of the original response distribution
- 3 = $50 < x \leq$ 75th percentile of the original response distribution
- 4 = $75 < x \leq$ 100th percentile of the original response distributiont

Binning allows for these scaled questions to be represented in a way similar to other survey items. A total of ninety-two survey questions were affected. Many items throughout the questionnaire contained instructions for participants to skip specific questions based on their responses to a certain question, e.g.,"If yes to question 1a, skip questions 1b and 1c." This was a source of a large amount of missing data and noise in our classification model. A new option, "0 = symptom not applicable", was introduced for these survey items. A total of sixty variables were affected.

3.3 Methodology

The steps of our proposed system are represented in Figure 1.

Fig. 1
SCHEMATIC OF PROPOSED SYSTEM

3.3.1 Machine learning algorithms

PASW® Modeler 14 software was used to create all of the prediction models described in this paper. We employ the CART algorithm in our decision tree classification models. A CART tree is a binary decision tree that is built by splitting a node into two child nodes repeatedly, starting with the root node that contains all of the training instances [25]. In addition to the CART algorithm, two additional algorithms are employed in an effort to improve the accuracy and generalizability of the prediction models [22].

- Bagging is a method that creates decision tree ensembles by generating replica training sets (bootstrap samples) from a standard training set. Trees are built on these replica training sets and predictions are combined via majority vote [26].
- Boosting is a method that builds decision trees sequentially while increasing the weights of the training samples that are misclassified by prior trees in the sequence. The tree predictions are combined in a scheme in which each tree is weighted according to its accuracy in classifying the training dataset [27].

3.3.2 Model validation

We partitioned our data into a rigorous 70% training set and 30% testing set as a means of validating our classification model.

3.3.3 Proposed schemes

1) Four-way single tree induction:

CFS vs. ICF vs. Exclusion vs. Control

In this paper, we implement a variety of ensemble methods to the classification of chronic fatigue syndrome. In order to make this comparison, we must first create a model that features a single CART decision tree and four-way classification as a basis for comparison. The decision tree algorithm is applied to the dataset, outputting each patient with one of the four possible labels. This scheme is represented in Figure 2.

.

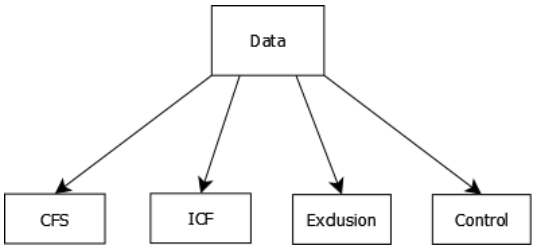


Fig. 2
A FOUR-WAY SINGLE DECISION TREE CLASSIFICATION

2) Two-way single tree induction:

CFS vs. Others
ICF vs. Others
Exclusion vs. Others
Control vs. Others

Building upon this idea of ensemble learning, we then look at four single decision tree models that classify between two categories. The decision tree algorithm is applied to the dataset, outputting each patient with one of two possible labels. For example, in the CFS vs. Others tree, a patient may be classified as "CFS" or "Others" (all categories that are not CFS). The same procedure is repeated with ICF vs. Others, Exclusion vs. Others, and Control vs. Others, with each tree outputting only one of two possible labels. This scheme is represented in Figure 3.

.

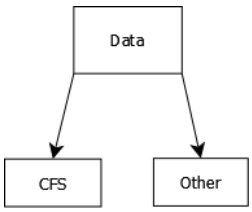


Fig. 3
A TWO-WAY SINGLE DECISION TREE CLASSIFICATION

3) Three-level decision tree ensemble:

ICF → CFS → Control → Exclusion
Exclusion → Control → CFS → ICF
Exclusion → CFS → ICF → Control
Control → ICF → CFS → Exclusion
Control → ICF → CFS → Exclusion (Bagging)
Control → ICF → CFS → Exclusion (Boosting)

This ensemble classifier couples the two-way decision trees described previously into a three-level ensemble architecture.
For example, in the ICF → CFS → Control → Exclusion ensemble, the first-level classifies an individual as "ICF" or "Others" (all categories that are not ICF). Those classified as "Others" move onto the next level of classification. The second-level classifies the first level "Others" as "CFS" or "Others" (all categories that are not CFS or ICF). Those classified as "Others" during this round move onto the last level of classification. The third and final layer is then used to classify second level "Others" as "Control" or "Others". Third level "Others" are assumed to be "Exclusion". The flow of this ensemble classifier is outlined in Figure 4. The same procedure is repeated for the combinations listed above.

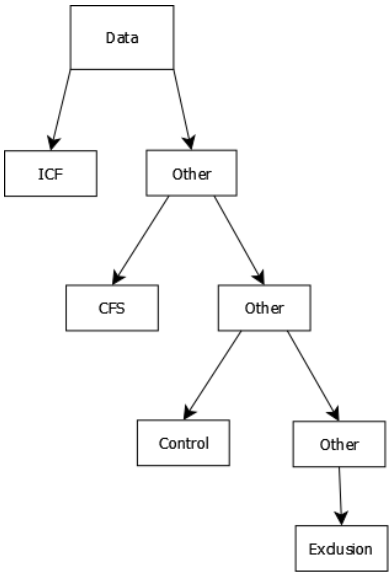


Fig. 4
A THREE-LEVEL ENSEMBLE CLASSIFIER

Rationale of level ordering in ensemble configurations:

- ICF → CFS → Control → Exclusion: In order of increasing sample size
- Exclusion → Control → CFS → ICF: In order of decreasing sample size
- Exclusion → CFS → ICF → Control: In order of increasing training set performance in the two-way single tree induction
- Control → ICF → CFS → Exclusion: In order of decreasing training set performance in the two-way single tree induction

- Control → ICF → CFS → Exclusion: In order of decreasing training set performance in the two-way single tree induction, with an additional boosting algorithm.
- Control → ICF → CFS → Exclusion: In order of decreasing training set performance in the two-way single tree induction, with an additional bagging algorithm.

4. EMPIRICAL EVALUATION

Table 1 contains the results from the four-way (CFS vs. ICF vs. Exclusion vs. Control) single decision tree classification. In this instance, the decision tree algorithm is applied to the dataset, outputting each patient with one of the four possible labels. The training and testing accuracies for this configuration serve as the baseline for this work as we explore and compare various ensemble methods. Table 1 additionally reports results from the single tree two-way classifications. In this set-up, the decision tree algorithm is applied to the dataset, outputting each patient with one of two possible labels. For example, in the CFS vs. Others tree, a patient may be classified as "CFS" or "Others" (all categories that are not CFS). The same procedure is repeated with ICF vs. Others, Exclusion vs. Others, and Control vs. Others, with each tree outputting only one of two possible labels. All two-way training and testing instances report better or approximately the same accuracies as the four-way classification. If one examines the relative difficulty of each classification problem, this appears to be a logical finding. Creating a model that only needs to distinguish between CFS and Others (not CFS) is a seemingly easier task than creating a model that needs to distinguish among four labels (CFS, ICF, Exclusion, or Control).

In summary, we have four two-way decision trees whose predictions we want to couple to create a multi-layer decision tree ensemble learner. We deduce that if the constituent trees of the ensemble learner have adequate accuracies, coupling them into an ensemble classifier will surely increase the generalizability of the model. Findings with ANN ensembles [18] report that the ensemble greatly outperforms single classifiers in the identification of cancerous lung cells. However, this lies contrary to what we found in our work. As shown in Table 2, while the training accuracies of the three-level tree ensemble are comparable to the two-way and four-way single tree methods, the ensemble testing accuracies are much lower than both two-way and four-way models. Bagging and boosting algorithms were applied to the three-level configuration with the best performance: Control → ICF → CFS → Exclusion. We encountered similar findings with these boosted and bagged results, comparable training accuracies but very poor testing accuracies, which means that the models failed to generalize well to new datasets.

While the ANN ensembles worked with biopsy data [18], we worked with medical survey data. As with any self-report measure, the medical survey introduces a great amount of bias in the dataset, more bias than would be present in a dataset with biological information. The principal weakness of this dataset is the large amount of missing data. While study participants were encouraged to complete all applicable questions in the survey, 28.2% of all values were still missing. With 108 cases and 486 features, missing nearly a third of all values is significant and likely to impact the accuracy of any prediction model created from it.

While decision trees are equipped to deal with datasets in which the number of variables is much larger than the number of samples, the variable-sample imbalance present in this dataset is likely introducing additional noise in our classification model. The dataset used in building classification models for rheumatoid arthritis and inflammatory bowel disease also suffered from this variable-sample imbalance, with several thousand features and a few hundred samples. However, their datasets, for the most part, were complete and did not rely on self-report methods [22]. Whether it is missing values, biased responses, variable-sample imbalance, or a combination of these issues, we were unable to enhance model generalizability with an ensemble classifier. The noisy dataset is likely to be accumulating error at each ensemble level and is ultimately producing a CFS classification model that generalizes very poorly. Efforts to reduce noise in the dataset, e.g., feature selection, filling in missing values, etc., may prove to be beneficial to the classification model.

Table 1
FOUR-WAY AND TWO-WAY SINGLE TREE PERFORMANCE

Classification order	Training accuracy(%)	Testing accuracy(%)
CFS vs. ICF vs. Exclusion vs. Control	85.53	71.88
CFS vs. Others	92.10	68.75
ICF vs. Others	100.0	90.63
Exclusion vs. Others	86.84	68.75
Control vs. Others	100.0	100.0

Table 2
THREE-LEVEL DECISION TREE ENSEMBLE PERFORMANCE

Classification order	Training accuracy(%)	Testing accuracy(%)
CFS vs. ICF vs. Exclusion vs. Control*	85.53	71.88
ICF → CFS → Control → Exclusion	92.11	62.50
Exclusion → Control → CFS → ICF	86.84	68.75
Exclusion → CFS → ICF → Control	84.21	62.50
Control → ICF → CFS → Exclusion	94.73	62.50

*Single four-way decision tree for comparison

Table 3
CONTROL → ICF → CFS → EXCLUSION ENSEMBLE PERFORMANCE

Method	Training accuracy(%)	Testing accuracy(%)
CFS vs. ICF vs. Exclusion vs. Control*	85.53	71.88
Control → ICF → CFS → Exclusion	94.73	62.50
Control → ICF → CFS → Exclusion (Bagging)	89.47	62.50
Control → ICF → CFS → Exclusion (Boosting)	97.37	65.63

*Single four-way decision tree for comparison

5. SIGNIFICANCE AND IMPACT

In this paper, we implemented a decision tree classifier on survey data to predict chronic fatigue syndrome. We began with the simplest model – a single decision tree that classified samples into one of four categories: CFS, ICF, Exclusion, or Control. This four-way decision tree classifier achieved modest predictive accuracy. We then examined the performance of two-way classification trees (CFS vs. Others,

ICF vs. Others, Exclusion vs. Others, Control vs. Others) and found that those generally outperformed the single four-way tree. We then coupled these four two-way decision trees into a hierarchical ensemble classifier comprised of three levels of classification. While we expected to improve classification accuracy of our model, we found that the ensemble approaches for all configurations performed poorly in comparison to the constituent two-way trees and the single four-way tree. Finally, we applied boosting and bagging algorithms to the ensemble classifier in an effort to improve classification accuracy. Both boosting and bagging algorithms worsened the predictive power of the ensemble classifier. This poor performance may be explained by the large amount of missing data, variable-sample imbalance, a subjective method of data retrieval (self-report), or any combination of these factors.

Existing methods in CFS classification have utilized non-linear classification methods like decision trees. The comprehensibility of decision trees make them valuable tools in the classification of complex diseases. The use of ensemble classifiers, a combination of single classifiers, has already been utilized to improve accuracy of disease prediction models. The creation of a robust classification model for chronic fatigue syndrome can help medical professionals better understand, diagnose, and treat this multi-factorial illness. Very little work has been done in creating a CFS machine learner from self-report data. Despite the results reported in this paper, the existing CFS prediction models [4], [5], [7], [8] hold promise that despite the results reported in this paper, a robust decision tree-based ensemble learner can be created from survey data.

6. ACKNOWLEDGMENTS

This work was supported by the National Science Foundation under Grant No. 0755407.

References

[1] Fukuda, K., Straus, S.E., Hickie, I., Sharpe, M.C., Dobbins, J.G., & Komaroff, A. (1994). The chronic fatigue syndrome: a comprehensive approach to its definition and study. *Annals of Internal Medicine*, 121(12), 953-59.

[2] Holmes, G.P., Kaplan, J.E., Gantz, N.M., Komaroff, A.L., Schonberger, L.B., et al. (1988). Chronic fatigue syndrome: a working case definition. *Annals of Internal Medicine*, 108(3), 387-89.

[3] Lofaro, D., Maestripieri, S., Greco, R., Papalia, T., Mancuso, D., et al. (2010). Prediction of chronic allograft nephropathy using classification trees. *Transplantation Proceedings*, 42(4), 1130-33.

[4] Hanson, S.J., Gause, W., & Natelson, B. (2001). Detection of immunologically significant factors for chronic fatigue syndrome using neural-network classifiers. *Clinical and Diagnostic Laboratory Immunology*, 8(3), 658-62.

[5] Linder, R., Dinser, R., Wagner, M., Krueger, G.R., & Hoffman, A. (2002). Generation of classification criteria for chronic fatigue syndrome using an artificial neural network and traditional criteria set. *In Vivo*, 16(1), 37-43.

[6] Hart, A., & Wyatt, J. (1990). Evaluating black-boxes as medical decision aids: issues arising from a study of neural networks. *Medical Informatics*, 15(3), 229-36.

[7] Huang, L.C., Hsu, S.Y., & Lin, E. (2009). A comparison of classification methods for predicting chronic fatigue syndrome based on genetic data. *Journal of Translational Medicine*, 7(1).

[8] Goertzel, B.N., Pennachin, C., De Souza Coelho, L., Maloney, E.M., Jones, J.F., & Gurbaxani, B. (2006). Allostatic load is associated with symptoms in chronic fatigue syndrome patients. *Pharmacogenomics*, 7(3), 485-94.

[9] McEwen, B.S., & Stellar, E. (1993). Mechanisms leading to disease. *Archives of Internal Medicine*, 153(18), 2093-101.

[10] Niu, Q., Huang, Z., Shi, Y., Wang, L., Pan, X., & Hu, C. (2010). Specific serum protein biomarkers of rheumatoid arthritis detected by MALDI-TOF-MS combined with magnetic beads. *International Immunology*, 22(7), 611-18.

[11] Foote, J.T. (1993). Decision tree probability modeling for HMM speech recognition. PhD Thesis, Brown University: Providence, RI.

[12] Greco, R., Papalia, T., Lofaro, D., Maestripieri, S., Mancuso, D., & Bonofiglio, R. (2010). Decisional trees in renal transplant follow-up. *Transplantation Proceedings*, 42(4), 1134-36.

[13] Ting, H.W., Wu, J.T., Chan, C.L., Lin, S.L., & Chen, M.H. (2010). Decision Model for Acute Appendicitis Treatment With Decision Tree Technology – A Modification of the Alvarado Scoring System. *Journal of the Chinese Medical Association*, 73(8), 401-406.

[14] Farion, K., Michalowski, W., Wilk, S., O'Sullivan, D., & Matwin, S. (2010). *Journal of Medical Systems*, 34(4), 551-562.

[15] Tsymbal, A., Pechenizkiy, M., & Cunningham, P. (2005). Diversity in search strategies for ensemble feature selection. *Information Fusion*, 6(1), 83-98.

[16] Tumer, K., & Ghosh, J. (1996). Error correlation and error reduction in ensemble classifiers. *Connection Science*, 8(3&4), 385-404.

[17] Abdel-Aal, R.E. (2005). Improved classification of medical data using abductive network committees trained on different feature subsets. *Computer Methods and Programs in Biomedicine*, 80(2), 141-53.

[18] Zhou, Z.H., Jiang, Y., Yang, Y.B., & Chen, S.F. (2002). Lung cancer cell identification based on artificial neural network ensembles. *Artificial Intelligence in Medicine*, 24(1), 25-36.

[19] Zhou, Z.H., & Jiang, Y. (2003). Medical diagnosis with C4.5 rule preceded by artificial neural network ensemble. *IEEE Transactions on Information Technology in Biomedicine*, 7(1), 37-42.

[20] Cunningham, P., Carney, J., & Jacob, S. (2000). Stability problems with artificial neural networks and the ensemble solution. *Artificial Intelligence in Medicine*, 20(3), 217-25.

[21] Sharkey, A.J.C., Sharkey, N.E., & Cross, S.S. (1998). Adapting an ensemble approach for the diagnosis of beast cancer. *Proceedings on the International Conference on Artificial Neural Networks*, 281-86.

[22] Geurts, P., Fillet, M., De Seny, D., Meuwis, M.A., Malaise, M., et al. (2005). Proteomic mass spectra classification using decision tree based ensemble methods. *Bioinformatics*, 21(15), 3138-45.

[23] Özcift, A. (2011). Random forests ensemble classifier trained with data resampling strategy to improve cardiac arrhythmia diagnosis. *Computers in Biology and Medicine*, 41(5), 265-271.

[24] Schietgat, L., Vens, C., Struyf, J., Blockeel, H., Kocev, D., & Džeroski, S. (2010). Predicting gene function using hierarchical multi-label decision tree ensembles. *BMC Bioinformatics*, 11(2).

[25] Breiman, L., Friedman, J.H., Olsen, R., & Stone, C.J. (1984). Classification and regression trees. CA: Wadsworth International.

[26] Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24, 123-40.

[27] Freund, Y., & Schapire, R. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55, 119-39.

Social Inclusion in Ambient Assisted Living Environments: Home Automation and Convenience Services for Elderly User

C. Röcker[1], M. Ziefle[1], and A. Holzinger[2]
[1] Human Technology Centre (HumTec), RWTH Aachen University
Theaterplatz 14, 52056 Aachen, Germany, {Roecker, Ziefle}@humtec.rwth-aachen.de
[2] Human-Computer Interaction, HCI4MED, Medical University Graz
Auenbruggerplatz 2/V, 8010 Graz, Austria, Andreas.Holzinger@meduni-graz.at

Abstract - *Traditionally, Ambient Assisted Living applications focus on health-related services, like the detection of emergency situations, long-term treatment of chronic diseases, or the prevention and early-detection of illnesses. Over the last years, more and more projects started to extend these classical healthcare scenarios by designing applications that explicitly aim at increasing well-being and social inclusion for elderly users. With the transition away from purely medical services towards integrated homecare environments, holistic design concepts and evaluation approaches will become necessary. This paper takes a detailed look at state-of-the-art applications in this field and illustrates emerging challenges for the design and development of future homecare systems.*

Keywords: Ambient Assisted Living, Ambient Intelligence, Ubiquitous and Pervasive Computing, Social Inclusion, Older Users.

1 Introduction

Extended life spans, declining birth rates, and the increase in single households contribute to a growing number of elderly people at risk for institutionalization [58]. But long-term institutionalization is not only a big financial burden to the healthcare system and therefore hard to maintain in the coming years, it is also not the preferred choice of many aging people. With the recent developments in the fields of information, communication and sensor technologies, solid technical infrastructures for providing new patient-centered home care solutions become available. While the majority of computer-supported health care tools designed in the last decades focused mainly on supporting care-givers and medical personnel, this trend recently changed with the introduction of assistive technology for providing supportive and adaptive services to ill or disabled individuals at home [15]. This is usually done by enhancing physical spaces with computers and sensing technology to make them sensitive and responsive to the presence of people [17][35]. The idea of such Ambient Assisted Living (AAL) environments is to provide assistive technologies for supporting people with specific demands in their daily activities and thereby allowing them to grow old in their own homes [22][25][43]. While Ambient Assisted Living environments can provide a multitude of personalized and context-adapted services in various areas of life, the majority of existing systems focuses on one of the three application domains: the detection of emergency situations, long-term treatment of chronic diseases, or the prevention and early-detection of illnesses. A detailed discussion of existing Ambient Assisted Living systems can be found in [50].

2 Safety and Well-Being in Ambient Assisted Living Environments

While older people are obviously in need of extended long-term care, they wish to maintain their independence as long as possible (see, e.g., [18],[28],[36],[42],[55] or [62] for more details). Studies show that many older people regard their home as a sanctuary and therefore prefer to stay at home, even at an increased risk to their health and safety [14]. This wish is often related to a perceived increase in the quality of life in a familiar environment. Generally, quality of life is a quite complex concept referring to the individual perception of one's "physical health, psychological state, level of independence, social relationships, personal beliefs and relationship to salient features in the environment" [60]. But as people age, their perceived quality of life is mostly determined by their ability to maintain an autonomous and independent life [61]. Hence, a variety of authors, including Bayer and Harper [4], Shafer [51], Starner et al. [56] and Mynatt et al. [44],

identified the loss of personal independence as a major concern of most elderly people. More and more Ambient Assisted Living projects address these needs by designing applications that explicitly aim at increasing safety and well-being for elderly users. The following section takes a closer look at state-of-the-art applications in this field.

3 Current Approaches Towards Social Inclusion

3.1 Intelligent User Interfaces

Several user studies (see, e.g., [54]) showed that elderly people and their families regard social inclusion, safety and home automation as important features of future homecare environments. With respect to the interaction in such environments, one of the main research challenges is the design of adequate user interfaces. This is due to the fact that elderly people vary considerably in their physical and cognitive abilities, which makes it difficult to use traditional interaction models [31]. Focusing on single interaction strategies may not always provide appropriate solutions [16] as many older computer users are affected by multiple disabilities, and such multiple minor (and sometimes major) impairments can interact. To address this problem various authors developed intelligent user interfaces, which support users according to their individual needs. For example, Gajos et al. [24] developed a system that allows an automatic generation of personalized user interfaces. The system assesses the user's motor abilities and automatically generates an interface adapted to the individual abilities of the user. Jung et al. [33] even developed a smart bed, which is able to 'sense' the intention of the user and act accordingly. Based on integrated pressure sensors, the bed is able to detect an intended movement of the user and automatically changes posture according to the user's intention [6].

There is also a considerable body of research on intelligent interfaces compensating for physical disabilities of users. For example, the *EyeMouse* interface [34][63] is a non-contact interaction device that enables physically impaired patients to control computers or robots by eye movement [6]. The *Input Adapter Tool* developed by Carter et al. [12] automatically modifies the interfaces of JAVA-based applications in order to improve the accessibility for users with restricted motor abilities. A similar system for web pages was developed by Mankoff et al. [40]. More artistic systems include *EyeDraw* [30], which creates paintings based on the eye movement of the user or *VoiceDraw* [26], which converts voice signals into paint strokes.

Another widely explored approach for supporting patient-device interaction are gesture interfaces. With the *Gesture Pendant*, Mynatt et al. [44] developed a wireless device, equipped with a camera and motion sensors, that enables users to control different services within smart home environments, like, e.g., closing the blinds, locking the door, or diming the lights, by using different hand gestures. In addition, the device is able to monitor the physical activity of its user and request help in case of an emergency. In contrast to the *Gesture Pendent*, the *Soft Remote Control* system [7][19][49] is integrated into the users' environment, which enables device-independent interaction between users and different services within the environment. The system allows users to control devices by pointing at an object within the room and using pre-defined hand gestures to execute different functions [5][8]. An alternative gesture recognition approach using vision systems is proposed by Jojic et al. [32]. Based on previous work by Kuno et al. [37], Brumitt and Cadiz [11] propose a multimodal home control system that combines gesture and speech input in order to provide natural interaction commands.

3.2 Connectedness-Oriented Communication

As people age and become less mobile, meeting other people outside the own home becomes more and more complicated if not impossible [41][42]. The resulting erosion of social networks is a natural consequence experienced by many older people [46], which is especially severe as social relationships are widely acknowledged as an important factor to well-being in old age [1][10][21][39]. This is also underlined by a survey conducted by Gabriel and Bowling [23] in the UK, which found that social relationships are the principal contributing factor to quality of life for people aged 65 and over. Hence, it is very important to provide older people with adequate information and communication technologies, that enable them to remain integrated in social life, despite of their age and existing disabilities [25][45].

Exchanging task-related information is only one aspect of communication. Especially within spatially distributed families, it is equally important to have informal interactions, spontaneous conversations and awareness of people and events at other sites [9]. Traditional communication technologies, like telephone and e-mail, focus mainly on content and are not adequate to support a feeling of connectedness. In contrast, connectedness-oriented communication is intended to support and augment social relationships among people by fostering a sense of connectedness among them [38]. The degree of connectedness is closely related to the 'social presence' a communication media is able to mediate [53]. The more a communication media is capable of mediating the personality and non-verbal symbols of a remote person, the higher is its social presence.

A broad variety of projects addressed this problem by designing ambient awareness systems, which make use of ambient display technologies to facilitate lightweight, informal and emotional forms of communication in smart environments. While ambient displays, by their nature, are not limited to certain kinds of data, most systems map presence information associated with other people to artefacts, situated or integrated in the environment. The systems are designed to support users in effortless maintaining awareness of the whereabouts and activities of others, and do so, by representing this information through changes in light, sound, movement, or temperature in the user's environment [59]. Looking at the design of the output interfaces for ambient awareness systems, two fundamentally different approaches are visible. One group of applications uses physical artefacts to visualize information, the other relies on large graphical displays showing abstract pictures compositions.

The usage of physical artefacts to represent awareness information is especially common in applications that focus on providing intimate affective communication between pairs of people. Strong and Gaver [57] developed several artefacts that enable pairs of people to stay in contact with each other. They envisioned mobile users to carry small picture frames whenever they are outside their home. When a user picks up the frame, information is transmitted to a stationary device at the partner's home. Depending on the device, a feather inside an acryl cone is whirled around (*Feather* artefact), or a heating element warms a container with essential oil (*Scent* artefact). Another tool for supporting intimate communication is the *Kiss Communicator* [3]. Blowing on the device creates ripples of light that can be sent to a remote partner's device, once a desirable pattern has been achieved. *Lumitouch* [13], *The Bench* [20], and *Shaker* [57] are other examples of tangible interactive display for supporting connectedness-oriented communication.

Another group of ambient awareness systems focuses on providing awareness in small, closely-related groups in home environments. The *FamilyPlanter* [38], for example, was developed to be used by family members living apart. The system uses infrared and ultrasonic sensors, to monitor the presence of a user, and transmits the status to a selected family member via a server. At the receiver's side the *FamilyPlanter* presents the received information by rotating and illuminating optical fibers. Besides automatically generated presence information, the system also offers a possibility to explicitly send 'messages' by touching a sensor at the *FamilyPlanter*, which causes the artefact at the receiver side to emit a sound. A similar approach is taken by Hindus et al. [29] with the *Lampshade Intentional Presence Lamp* (IPL), a decorative artefact that serves both as a lamp and a communication device. In contrast to the *FamilyPlanter*, the presence of people is not automatically captured. Instead, presence information is only communicated, if the user explicitly activates the *Lampshade* device. Other system supporting awareness in small intimate groups are *Digital Family Portraits* [45], *Gleams of People* [48], *Faint-Pop* [47], or *Curtain IPL* [29].

4 New Challenges for the Design and Evaluation of Future Systems

With the transition away from purely medical services towards integrated homecare environments, holistic design concepts and evaluation approaches will become more and more important. For the conceptualization, design and evaluation of future homecare systems it is not sufficient anymore to rely on one prototypical user, one prototypical usage situation, and a one-to-one-relation of technology and task context [51]. Individuals live in environments and expect technology to be adaptive and useful, however, what is judged as useful may change depending on the application domain and usage context. For example, it has been shown [2] that one and the same mobile device can evoke different perceived benefits and drawbacks when used in a communication scenario compared to a medical monitoring scenario. Hence, the acceptance or rejection of a technical product is neither static nor independent from the specific context, in which the technology is used.

For a long time, design aspects have been discussed from a predominantly functional perspective. With the increasing penetration of technology into private spheres, technology must comply with the needs and wants of a diverse user group in order to be successful and fully accepted by those for which technology is designed. Evidently, a system or service can be usable in terms of performance measures, but, at the same time, it can be completely rejected. Therefore, the design focus should not only be on efficiency, task-appropriateness, and ease-of-use. A number of studies show that users desire more than pure functionality, but prefer interfaces with a high social and hedonic value, as they provide stimulation, identity, and valuable memories (see, e.g. [27] or [64]). Hence, other dimensions of usage should be included, like the pleasure of using a device or its appropriateness for a specific usage setting. Consequently, the design of future homecare environments should follow a strictly user driven approach starting by analyzing the relationship of users and technological product.

5 References

[1] Adams, R. G., Blieszner, R. (1995). Aging Well with Friends and Family. In: *American Behavioural Scientist*, Vol. 39, No. 2, pp. 209 - 224.

[2] Arning, K., Gaul, S., Ziefle, M. (2010). Same Same but Different - How Service Contexts of Mobile Technologies Shape Usage Motives and Barriers. In: G. Leitner, M. Hitz, A. Holzinger (Eds): *HCI in Work & Learning, Life & Leisure, 6th Symposium of the WG HCI&UE of the Austrian Computer Society* (USAB'10). Lecture Notes in Computer Science 6389. Springer, Berlin, Germany, pp. 34 - 54.

[3] Barley, N. (1999). Lost and Found: Critical Voices in New British Design. Birkhäuser, Berlin, Germany.

[4] Bayer, A.-H., Harper, L. (2000). Fixing to Stay: A National Survey on Housing and Home Modification Issues - Executive Summary. American Association of Retired Persons.

[5] Bien, Z. Z., Lee, H. E., Do, J.-H., Kim, Y.-H., Park, K.-H., Yang, S.-E. (2008). Intelligent Interaction for Human-Friendly Service Robot in Smart House Environment. In: *International Journal of Computational Intelligence Systems*, Vol. 1, No. 1, pp. 77 - 93

[6] Bien, Z. Z., Park, K.-H., Kim, D.-J., Jung, J.-W. (2004). Welfare-Oriented Service Robotic Systems: Intelligent Sweet Home & KARES II. In: Z. Z. Bien, D. H. Stefanov (Eds.): *Advances in Rehabilitation Robotics: Human-Friendly Technologies on Movement Assistance and Restoration for People with Disabilities*, Lecture Notes in Control and Information Sciences, Springer, Berlin, Germany, pp. 57 - 94.

[7] Bien, Z. Z., Park, K.-H., Kim, J. B., Do, J.-H., Stefanov, D. H. (2003). User-Friendly Interaction/Interface Control of Intelligent Home for Movement-Disabled People. In: *Proceedings of the 10th International Conference on Human-Computer Interaction*, Grete, Greece.

[8] Bien, Z. Z., Park, K.H., Bang, W.-C., Stefanov, D. H. (2002). LARES: An Intelligent Sweet Home for Assisting the Elderly and the Handicapped in Universal Access and Assistive Technology. In: S. Keates, P. Langdon, P. J. Clarkson, P. Robinson (Eds.): *Proceedings of the Workshop on Universal Access and Assistive Technology* (CWUAAT'02), Cambridge, UK, March 25 - 27, Springer, Heidelberg, Germany, pp. 43 - 46.

[9] Bly, S., Harrison, S. R., Irwin, S. (1993). Media Spaces: Bringing People Together in a Video, Audio, and Computing Environment. In: *Communications of the ACM*, Vol. 36, No. 1, pp. 28 - 46.

[10] Borglin, G., Edberg, A., Hallberg, I. R. (2005). The Experience of Quality of Life Among Older People. In: *Journal of Aging Studies*, Vol. 19, No. 2, pp. 201 - 220.

[11] Brumitt, B., Cadiz, J. J. (2001). Let There Be Light - Examining Interfaces for Homes of the Future. In: *Proceedings of IFIP Conference on Human-Computer Interaction* (INTERACT'01), Tokyo, Japan, pp. 375 - 382.

[12] Carter, S., Hurst, A., Mankoff, J., Li, J. (2006). Dynamically Adapting GUIs to Diverse Input Devices. In: *Proceedings of the international ACM SIGACCESS Conference on Computers and Accessibility* (ASSETS'06), ACM Press, New York, NY, USA, pp. 63 - 70.

[13] Chang, A., Resner, B., Koerner, B., Wang, X., Ishii, H. (2001). LumiTouch: An Emotional Communication Device. In: *Proceedings of the Conference on Human Factors in Computing Systems* (CHI'01), pp. 313 - 314.

[14] Cook, D. J., Das, S. K. (2007). How Smart are Our Environments? An Updated Look at the State of the Art. In: *Journal of Pervasive and Mobile Computing*, Vol. 3, No. 2, pp. 53 - 73.

[15] Cortés, U., Annicchiarico, R., Urdiales, C., Barrué, C., Martínez, A., Villar, A., Caltagirone, C. (2007). Supported Human Autonomy for Recovery and Enhancemnt of Cognitive and Motor Abilities Using Agent Technologies. In: R. Annicchiarico, U. Cortès, C. Urdiales (Eds.): *Agent Technology and E-Health.* Birkhäuser, Basel, Switzerland, pp. 117 - 140.

[16] De Hert, P., Mantovani, E. (2008). *Intelligent User Interface*. Working Document of the Project "SENIOR: Social Ethical and Privacy Needs in ICT for Older Citizens: A Dialogue Roadmap" (IST 216820).

[17] de Ruyter, B., Pelgrim, E. (2007). Ambient Assisted-Living Research in CareLab. In: *ACM Interactions*, Vol. 14, No. 4, pp. 30 - 33.

[18] Dewsbury, G., Edge, M. (2000). Designing the Home to Meet the Needs of Tomorrow … Today: Deconstructing and Rebuilding the Home for Life. In: *Proceedings of the International Conference of the European Network for Housing Research, Housing in the 21st Century: Fragmentation and Re-Orientation*, CD-ROM.

[19] Do, J.-H., Kim, J.-B., Park, K.-H., Bang, W.-C., Bien, Z. Z. (2002). Soft Remote Control System Using Hand Pointing Gesture. In: *International Journal of Human-Friendly Welfare Robotic Systems*, Vol. 3, No. 1, pp. 27 - 30.

[20] Dunne, A., Raby, F. (1995). Fields and Thresholds. In: *Architectural Design*, Vol. 65, No. 11-12, pp. 60 - 65.

[21] Fiori, K. L., Antonucci, T. C., Cortina, K. S. (2006). Social Network Typologies and Mental Health Among Older Adults. In: *The Journal of Gerontology: Psychological Science*s, Vol. 61, No.1, pp. 25 - 32.

[22] Fuchsberger, V. (2008). Ambient Assisted Living: Elderly People's Needs and How to Face Them. In: *Proceeding of the ACM International Workshop on Semantic Ambient Media Experiences* (SAME'08), October 31, 2008, Vancouver, British Columbia, Canada, pp. 21- 24.

[23] Gabriel, Z., Bowling, A. (2004). Quality of Life in Old Age From the Perspectives of Older People. In: A. Walker, C. H. Hennessy (Eds.): *Growing Older: Quality of Life in Old Age*, Open University Press, Buckingham, PA, USA, pp. 14 - 34.

[24] Gajos, K. Z., Wobbrock, J. O., Weld, D. S. (2008). Improving the Performance of Motor-Impaired Users with Automatically-Generated, Ability-Based Interfaces. In: *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems* (CHI'08), ACM Press, pp. 1257 - 1266.

[25] Hanak, D., Szijarto, G., Takacs, B. (2007). A Mobile Approach to Ambient Assisted Living. In: *Proceedings of the IADIS Multi Conference on Computer Science and Information Systems 2007* (MCCSIS'07), July 3 – 8, Lisbon, Portugal.

[26] Harada, S., Wobbrock, J. O., Landay, J. A. (2007). Voicedraw: A Voice-Driven Hands-Free Drawing Application. In: *Proceedings of the international ACM SIGACCESS Conference on Computers and Accessibility* (ASSETS'07), ACM Press, pp. 27 - 34.

[27] Hassenzahl, M. (2010). *Experience Design – Technology for All the Right Reasons.* Morgan & Claypool, San Rafael, CA, USA.

[28] Heinze, R. G., Eichener, V., Naegele, G., Bucksteg, M., Schauerte, M. (1997). Neue Wohnung auch im Alter – Folgerungen aus dem demographischen Wandel für Wohnungspolitik und Wohnungswirtschaft. Schader-Stiftung, Darmstadt, Germany.

[29] Hindus, D., Mainwaring, S., Leduc, N., Hagström, A. E., Bayley, O. (2001). Casablanca: Designing Social Communication Devices for the Home. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI'01), ACM Press, pp. 325 - 332.

[30] Hornof, A., Cavender, A., Hoselton, R. (2004). Eyedraw: A system for Drawing Pictures with Eye Movements. In: *Proceedings of the International ACM SIGACCESS Conference on Computers and Accessibility* (ASSETS'04), ACM Press, pp. 86 - 93.

[31] Jacobs, A., Pierson, J. (2007). Walking the Interface: Domestication Reconsidered? In: J. Cornelis, M. Wynants (Eds.): *Brave New Interfaces*, VUB Press, Brussels, Belgium, pp. 205 - 215.

[32] Jojic, N., Brumitt, B., Meyers, B. Harris, S., Huang, T. (2000). Detection and Estimation of Pointing Gestures in Dense Disparity Maps. In: Proceedings of the Forth International Conference on Automatic Face and Gesture Recognition, pp. 468 - 475.

[33] Jung, J. W., Lee, C. Y., Lee, J. J., Bien, Z. Z. (2003). User Intention Recognition for Intelligent Bed Robot System. In: Proceedings of the Eighth International Conference on Rehabilitation Robotics (ICORR'03), pp 100 - 103.

[34] Kim, D. H., Kim, J. H., Chung, M. J. (2001). A Computer Interface for the Disabled Using Eye-Gaze Information. In: *International Journal of Human-Friendly Welfare Robotic Systems*, Vol. 2, No. 3, pp. 22 - 27.

[35] Knaup-Gregori, P., Ecker, M., Albashiti, F., Spitalewsky, K., Dickhaus, H. (2009). Individualisierte Gesundheitsportale als zentrale IT-Komponenten für Ambient Assisted Living: Motivation und Anforderungen. In*: Proceedings of the Second German Congress on Ambient Assisted Living*, January 27 - 28, 2009, Berlin, Germany. VDE, Berlin, Germany. CD-ROM.

[36] Kühne, H., Behner, U., Henke, S., Nelles, S. (2009). Das selbstbestimmte Leben älterer chronisch Kranker und Risiko-patienten im eigenen Zuhause erfordert innovative und bezahlbare Versorgungslösungen. In: *Proceedings of the Second German Congress on Ambient Assisted Living*, January 27 - 28, 2009, Berlin, Germany. VDE, Berlin, Germany. CD-ROM.

[37] Kuno, Y., Ishiyama, T., Nakanishi, S., Shirai, Y. (1999). Combining Observations of Intentional and Unintentional Behaviors for Human-Computer Interaction. In: *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI'99), ACM Press, pp. 238 - 245.

[38] Kuwabara, K., Watanabe, T., Ohguro, T., Itoh, Y., Maeda, Y. (2002) Connectedness Oriented Communication: Fostering a Sense of Connectedness to Augment Social Relationships. In: *Journal of the Information Processing Society of Japan*, Vol. 43, No. 11, pp. 3270 - 3279.

[39] Lindley, S. E., Harper, R., Sellen, A. (2008). Designing for Elders: Exploring the Complexity of Relationships in Later Life. In*: Proceedings of the 22nd British HCI Group Annual Conference on HCI 2008: People and Computers XXII: Culture, Creativity, Interaction - Volume 1*, September 1 - 5, Liverpool, UK, pp. 77-86.

[40] Mankoff, J., Dey, A., Batra, U., Moore, M. (2002). Web Accessibility for Low Bandwidth Input. In: *Proceedings of the International ACM Conference on Computers and Accessibility* (ASSETS'02), ACM Press, New York, NY, USA, pp. 17 - 24.

[41] Marshall, M. (2000). Astrid: A Social & Technological Response to Meeting the Needs of Individuals with Dementia and their Carers, Hawker Publications, London, UK.

[42] Meyer, S. (2009). Der Nutzer im Zentrum: Anforderungen, Wünsche, Erfahrungen der älteren Senioren. In: *Proceedings of the Second German Congress on Ambient Assisted Living*, January 27 - 28, 2009, Berlin, Germany. VDE, Berlin, Germany. CD-ROM.

[43] Mukasa, K. S., Holzinger, A., Karshmer, A. I. (2008). Workshop on Intelligent User Interfaces for Ambient Assisted Living. In: *Proceedings of International Conference on Intelligent User Interfaces* (IUI'08), January 13 - 16, Maspalomas, Gran Canaria, Spain, p. 436.

[44] Mynatt, E. D., Melenhorst, A.-S., Fisk, A. D., Rogers, W. A. (2004). Aware Technologies for Aging in Place: Understanding User Needs and Attitudes. In: *Pervasive Computing*, Vol. 3, No. 2, pp. 36 - 41.

[45] Mynatt, E. D., Rowan, J., Jacobs, A., Craighill, S. (2001). Digital Family Portraits: Supporting Peace of Mind for Extended Family Members. In*: Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI'01), pp. 333 - 340.

[46] Nehmer, J., Becker, M., Karshmer, A., Lamm, R. (2006). Living Assistance Systems: An Ambient Intelligence Approach. In: *Proceeding of the International Conference on Software Engineering* (ICSE '06), May 20 - 28, Shanghai, China, ACM Press, New York, NY, pp. 43 - 50.

[47] Ohguro, T., Kuwabara, K., Owada, T., Shirai, Y. (2001) FaintPop: In Touch with the Social Relationships. In: *Proceedings of the Joint JSAI Workshop on New Frontiers in Artificial Intelligence*, pp. 11 - 18.

[48] Ohguro, T., Yoshida, S., Kuwabara, K. (1999). Gleams of People: Monitoring the Presence of People with Multi-Agent Architecture. In: *Proceedings of the Second Pacific Rim International Workshop on Multi-Agents* (PRIMA'99), pp. 170 - 182.

[49] Park, K. H., Bien, Z. Z. (2003). Intelligent Sweet Home for Assisting the Elderly and the Handicapped. In: *Proceedings of the 1st International Conference on Smart Homes and Health Telematics* (ICOST'03), Paris, France, pp. 151 - 158.

[50] Röcker, C. (2011). Smart Medical Services: A Discussion of State-of-The-Art Approaches. In: S. Thatcher (Ed.): *Proceedings of the International IEEE Conference on Machine Learning and Computing* (ICMLC 2011), Volume 1, February 26-28, 2011, Singapore, IEEE Press, pp. 334 - 338.

[51] Röcker, C., Feith, A. (2009). Revisiting Privacy in Smart Spaces: Social and Architectural Aspects of Privacy in Technology-Enhanced Environments. In: *Proceedings of the International Symposium on Computing, Communication and Control* (ISCCC'09), October 9 -11, 2009, Singapore, pp. 201 - 205.

[52] Shafer, R. (2000). *Housing America's Seniors.* Joint Center for Housing Studies, Harvard University, Cambridge, MA.

[53] Short, J., Williams, E., Christie, B. (1976). *The Social Psychology of Telecommunications*. John Wiley and Sons, London, UK.

[54] SOPRANO (2007). *SOPRANO Requirements Specification.* Deliverable D1.2.2 of the project "Service-Oriented Programmable Smart Environments for Older Europeans" (SOPRANO).

[55] Spellerberg, A., Schelisch, L. (2009). Ein dreiviertel Jahr mit PAUL: Assisted Living in Kaiserslautern. In: *Proceedings of the Second German Congress on Ambient Assisted Living*, January 27 - 28, Berlin, Germany. VDE, Berlin, Germany. CD-ROM.

[56] Starner, T., Auxier, J., Ashbrook, D., Gandy, M. (2000). The Gesture Pendant: A Self-Illuminating, Wearable, Infared Computer Vision System for Home Automation Control and Medical Monitoring. In: *Proceedings of the IEEE International Symposium on Wearable Computing* (ISWC'00), IEEE Press, pp. 87 - 94.

[57] Strong, R., Gaver, B. (1996). Feather, Scent, and Shaker: Supporting Simple Intimacy. In: *Poster Proceedings of the ACM Conference on Computer Supported Cooperative Work* (CSCW'96), p. 444.

[58] Vergados, D., Alevizos, A., Mariolis, A., Caragiozidis, M. (2008). Intelligent Services for Assisting Independent Living of Elderly People at Home. In: *Proceedings of the International ACM Conference on Pervasive Technologies Related to Assistive Environments* (PETRA'08), July 15 - 19, 2008, Athens, Greece. CD-ROM.

[59] Washington, W. (2001). *Exploring Ambient Media Presence Awareness.* Masters Degree Project Report. Department of Technical Communication, University of Washington, Seattle, Washington, USA.

[60] WHO (1994). *Statement Developed by WHO Quality of Life Working Group.* WHO Health Promotion Glossary 1998, WHO/HPR/HEP/98.1, World Health Organization, Geneva, Switzerland.

[61] WHO (2002). *Active Aging: A Policy Framework.* World Health Organization, Geneva, Switzerland.

[62] Wichert, R., Gaugisch, P., Norgall, T., Becker, M. (2009). Individuelle Gestaltung und Anpassung bestehender Wohnkonzepte. In: *Proceedings of the Second German Congress on Ambient Assisted Living*, January 27 - 28, 2009, Berlin, Germany. VDE, Berlin, Germany. CD-ROM.

[63] Yoo, D. H., Chung, M. J. (2002). Vision-Based Eye Gaze Estimation System Using Robust Pupil Detection and Corneal Reflections. In: *International Journal of Human-Friendly Welfare Robotic Systems*, Vol. 3, No. 4, pp 2 - 8.

[64] Ziefle, M., Jakobs, E.-M. (2010). New Challenges in Human Computer Interaction: Strategic Directions and Interdisciplinary Trends. In: *Proceedings of the International Conference on Competitive Manufacturing Technologies* (COMA'10), University of Stellenbosch, South Africa, pp. 389 - 398.

A Hybrid Adaptive Multi Sensor Data Fusion for Estimation of Skeletal Muscle Force for Prosthetic Hand Control

Parmod Kumar, Chandrasekhar Potluri, Anish Sebastian, Yimesker Yihun, Adnan Ilyas, Madhavi Anugolu, Rohit Sharma, Steve Chiu, Jim Creelman, Alex Urfer, D. Subbaram Naidu, *Fellow IEEE,* and Marco P. Schoen, *Senior Member IEEE*

Abstract – ***Effective use of upper extremity prostheses depends on the two critical aspects of precise position and force control. Surface electromyographic (sEMG) signals can be used as a control input for the position and force actions related to the prosthesis. In this paper, we use the measured sEMG signals to estimate skeletal muscle force. Further, we consider skeletal muscle as a system and System Identification (SI) is used to model multi-sensor sEMG and skeletal muscle force. The sEMG signals are filtered utilizing optimized nonlinear Half-Gaussian Bayesian filter, and a Chebyshev type-II filter provides the muscle force signal. The filter optimization is accomplished using a Genetic Algorithm (GA). Multi- linear and nonlinear models are obtained with sEMG data as input and skeletal muscle force of a healthy human hand as an output for three sensors. The outputs of these models for three sensors are fused with a probabilistic Kullback Information Criterion (KIC) for model selection and an adaptive probability of KIC. The final fusion based force for multi-sensor sEMG gives improved estimate of the skeletal muscle force.***

Keywords: *sEMG, SI, GA, KIC*

I. INTRODUCTION

In the last decade there have been active research efforts in the field of prosthetics. According to [1] the United States has 1.7 million of amputees and this figure is on the rise since 2003 due to the wars in Afghanistan and Iraq [2].

Manuscript received March 31, 2011. This work was supported in part by the US Department of the Army, under the award number W81XWH-10-1-0128.

Parmod Kumar is with Measurement and Control Engineering Research Center (MCERC), School of Engineering, Idaho State University, Pocatello, Idaho 83209, USA (email: kumaparm@isu.edu).

Chandrasekhar Potluri, Anish Sebastian, Yimesker Yihun, Adnan Ilyas, Madhavi Anugolu, and Rohit Sharma are with MCERC, School of Engineering, Idaho State University, Pocatello, Idaho 83209, USA (e-mail: potlchan@isu.edu, sebaanis@isu.edu, yihuyime@isu.edu, ilyaadna@isu.edu, anugmadh@isu.edu, sharrohi@isu.edu).

Steve Chiu is with Department of Electrical Engineering and Computer Science, MCERC, Idaho State University, Pocatello, Idaho 83201 USA (email: chiustev@isu.edu).

Jim Creelman and Alex Urfer are with Department of Physical and Occupational Therapy, Idaho State University, Pocatello, Idaho 83209, USA (email: creejame@isu.edu, urfealex@isu.edu).

D. Subbaram Naidu is with Department of Electrical Engineering and Computer Science, MCERC, Idaho State University, Pocatello, Idaho 83201 USA (email: naiduds@isu.edu).

Marco P. Schoen is with Department of Mechanical Engineering, MCERC, Idaho State University, Pocatello, Idaho 83209, USA (email: schomarc@isu.edu).

Individuals with upper extremity amputations not only have a reduction in function, but also struggle with numerous psychological issues which may further complicate the appropriate control and use of the prosthesis. Previous research on the prosthetic hand design was mainly based on the electromyography (EMG) signals [3-4]. The EMG signal recorded at the surface of the limb is known as surface EMG (sEMG). The sEMG is an electric voltage ranging between -5 and +5 mV, which reflects the electrical activity associated with voluntary muscle contraction and which has been used to control the position and force of the hand prosthesis [5-6]. The skeletal muscle force and the sEMG signals are related and higher sEMG amplitude generally correlates with greater force production. However, there are various other factors that affect this relationship. The sEMG signals are random, complex and dynamic in nature because of varying motor unit recruitment, crosstalk, and biochemical interaction within the muscular fibers. Hence all these factors contribute towards the nonlinear relationship between the two.

Since sEMG is the control input for the prosthesis, precise control and correct interpretation of the measured sEMG signal is important. In previous works, the authors have published similar work using multi sensor data fusion and Output Error (OE) model and modal transformation [7], and multi sensor data fusion using nonlinear models [8].

The present work is an extension of our previous work [9] where we proposed estimation of the skeletal muscle force using an adaptive data fusion algorithm with hybrid multi linear and nonlinear models. In [9] the sEMG data was based only from measuring the motor point sEMG signal, whereas in the present work we extend this to three sensors for the ring finger of the dominant hand of a healthy male subject. Figure 1 presents the flow chart of the work done in this paper. The sEMG signals are acquired from the forearm of a healthy subject and filtered utilizing Genetic Algorithm (GA) based optimized nonlinear Half-Gaussian Bayesian filter. On the other hand, the skeletal muscle force signal is filtered using a Chebyshev type-II filter. The sEMG is considered as input whereas force signal is considered as an output. Since each model captures the dynamics differently, we obtained multiple linear and nonlinear models using System Identification (SI) for the sEMG data corresponding to the three sensors. First, the outputs of all the linear and nonlinear models from the three sensors are fused separately and then

the resultant three outputs are fused with a probabilistic Kullback Information Criterion (KIC) for model selection and an adaptive probability of KIC. The final fused output using this approach gives improved skeletal muscle force estimates.

The paper is organized as follows, the present section covers the introduction and literature work, which is followed by the description of the experimental set-up and pre-processing, then system identification and linear and nonlinear modeling is discussed, followed by the data fusion algorithm. The paper is concluded by a section with, results and discussions, and followed by the conclusions and future work.

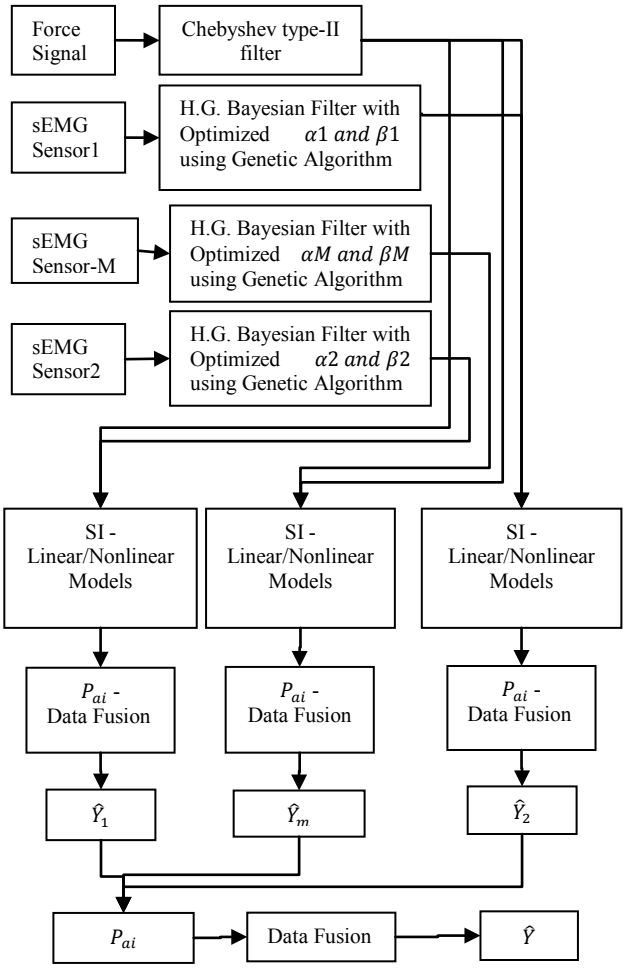


Figure 1. Flow chart for estimation of skeletal muscle force using an adaptive data fusion algorithm.

II. Experimental Set-Up and Pre-Processing

The sEMG and skeletal muscle force signals are acquired using LabVIEW™ 8.2 simultaneously at a sampling rate of 2000 Hz. A DELSYS® Bagnoli-16 EMG system with DE-2.1 differential EMG sensors is used for sEMG data capturing. However, the corresponding force data was captured using NI ELVIS with Interlink Electronics FSR 0.5" circular force sensor. The experimental set-up is shown in Figure 2. One sEMG sensor was placed on the motor point of the ring finger and two adjacent to the motor point of a healthy subject. Skin surface on the forearm of the subject was prepared with the International Society of Electrophysiology and Kinesiology (ISEK) protocol prior to placing sEMG sensors.

The nonlinear Bayesian filter yields good sEMG data and significantly reduces the influence of noise and it also extracts a signal that best describes the EMG signals and may permit effective use in prosthetic control [10]. The sEMG signal can be modeled with the assumption that the conditional probability of the rectified EMG signal is a filtered random process with random rate. The "Half-Gaussian measurement model" for the rectified EMG signal is given by Equation (1), [10].

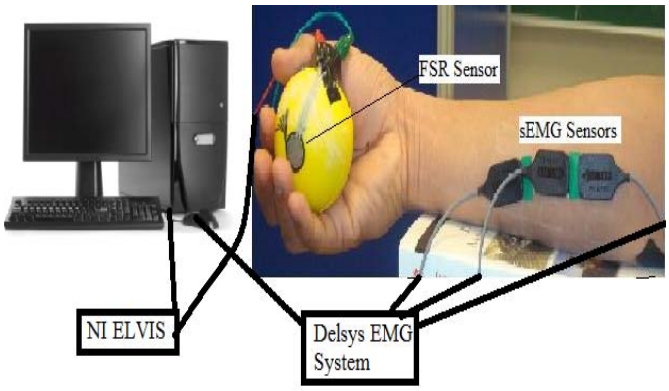


Figure 2. Experimental Set-Up.

$$P(emg|x) = 2 * exp(-\frac{emg^2}{2*x^2})/(2 * \pi * x^2)^{1/2}. \qquad (1)$$

The likelihood function for the rate evolving in time is given by the Fokker–Planck partial differential equation, where the two free parameters in the signal are the expected rate of gradual drift, α and the expected rate of sudden shifts β [10]. The two free parameters of the sEMG signal are optimized using an elitism based GA. On the other hand the skeletal muscle force signal is filtered with a Chebyshev type II low pass filter.

III. System Identification – Linear and Nonlinear Modeling

A system can be modeled either by using physics based mathematical models or by System Identification (SI) [11]. Parametric SI gives a dynamic model such as the Auto Regressive with eXogenous input (ARX) or Output Error (OE) of the system in mathematical form. In this research, we use the SI approach to model the two signals. The sEMG is the input to the system and the intended finger/hand force is output. In this work, mutli- linear and nonlinear models are obtained for modeling of sEMG and hand/finger force signals for three sensors for the ring finger of the dominant hand of a healthy subject. Six linear and three nonlinear models are obtained for the motor point sEMG signal, three linear and five nonlinear models are obtained for the ring1 sensor, and six linear and five nonlinear models are obtained for ring2

sensor data. Table I and II gives the structures of linear and nonlinear models respectively.

The model order of the various models used in this work are as follows: linear models for the motor point data, OE model of order 6, ARX model of order 6, ARMAX model of model order 5, Box-Jenkins (BJ) model of order 18, State-Space model with subspace method (N4SID) of order 12 and a State-Space model with prediction error/maximum likelihood method (PEM) of order 12 are obtained using SI. For ring1 sensor (sensor adjacent to the motor point sensor) data OE model of order 17, ARMAX model of order 35 and BJ model of order 11 are obtained. For ring2 sensor (sensor on opposite side of ring1 sensor) data OE model of order 45, ARX model of order 12, ARMAX model of order 17, BJ model of order 14, N4SID model of order 13 and PEM model of order 22 are obtained. Table I gives the structure of all the linear models. In table I y is output, t is time, $B(q)$, $F(q)$, $A(q)$, $C(q)$, $D(q)$, are polynomials, q is a backward shift operator, u is input, n_k is delay and e is error [12]. For State-Space models:- x is state, t is time, Ts is sampling time, u is input, e is error, A, B, K, C, and D are system matrices, and y is output [12].

TABLE I
LINEAR MODELS AND THEIR STRUCTURES

Linear Model Name	Linear Model Structure
Output Error	$y(t) = \frac{B(q)}{F(q)}u(t-n_k) + e(t)$
ARX	$A(q)y(t) = B(q)u(t-n_k) + e(t)$
ARMAX	$A(q)y(t) = B(q)u(t-n_k) + C(q)e(t)$
Box-Jenkins	$y(t) = \frac{B(q)}{F(q)}u(t-n_k) + \frac{C(q)}{D(q)}e(t)$
State-Space – subspace method	$x(t+Ts) = Ax(t) + Bu(t) + Ke(t)$ $y(t) = Cx(t) + Du(t) + e(t)$
State-Space – prediction error/maximum likelihood method	$x(t+Ts) = Ax(t) + Bu(t) + Ke(t)$ $y(t) = Cx(t) + Du(t) + e(t)$

Table II gives the structures of nonlinear models. The nonlinear models for the motor point data are obtained as, the nonlinear ARX with a sigmoidnet nonlinearity estimator, nonlinear Wiener-Hammerstein models with a piecewise linear nonlinearity estimator, and a saturation based estimator. For each ring1 and ring2 signal data sets, five nonlinear Wiener-Hammerstein models with nonlinearity estimators as 'piecewise linear – pwlinear,' 'sigmoidnet,' 'saturation,' 'deadzone,' and wavelet network' are obtained.

In table II f is a nonlinear function for the nonlinear ARX model. Inputs to f are model regressors. $y_p(t)$ is the predicted output as a weighted sum of past output values and current and past input values. For the nonlinear Hammerstein-Wiener model $u(t)$ and $y(t)$ are the inputs and outputs for the system, respectively. f and h are nonlinear functions that corresponds to the input and output nonlinearity, respectively. $w(t)$ and $x(t)$ are internal variables. $w(t)$ has the same dimension as $u(t)$. $x(t)$ has the same dimension as $y(t)$. $B(q)$ and $F(q)$ in the linear dynamic block are polynomials in the backward shift operator. For nonlinear Hammerstein-Weiner model with deadzone nonlinearity estimator, F is a nonlinear function of x with the properties given in the table II. For nonlinear Hammerstein-Weiner model with wavelet network as nonlinearity estimator, $\kappa(s)$ as a wavelet function, and β_k is a row vector such that $\beta_k(x-\gamma_k)$ is a scalar. If only the input nonlinearity is present, the model is called the Hammerstein model. If only the output nonlinearity is present, the model is called the Wiener model [12].

TABLE II
NONLINEAR MODELS AND THEIR STRUCTURES

Nonlinear Model Name	Nonlinear Model Structure
Nonlinear ARX – Sigmoidnet	$y_p(t) = f(y(t-1), y(t-2), y(t-3), \dots, u(t-1), u(t-2), \dots)$; $f(z) = \frac{1}{e^{-z}+1}$ is sigmoid function.
Nonlinear HW – Piecewise Linear – pwlinear	$w(t) = f(u(t))$, $x(t) = \frac{B_{ji}(q)}{F_{ji}(q)}w(t)$, $y(t) = h(x(t))$;
Nonlinear HW – Sigmoidnet	$y_p(t) = f(y(t-1), y(t-2), y(t-3), \dots, u(t-1), u(t-2), \dots)$; $f(z) = \frac{1}{e^{-z}+1}$ is sigmoid function.
Nonlinear HW – Saturation	$w(t) = f(u(t))$, $x(t) = \frac{B_{ji}(q)}{F_{ji}(q)}w(t)$, $y(t) = h(x(t))$;
Nonlinear HW – Deadzone	$y = F(x)$; $a \le x < b, \quad F(x) = 0$ $x < a, \quad F(x) = x - a$ $x \ge b, \quad F(x) = x - b$
Nonlinear HW – Wavelet Network	$g(x) = \sum_{k=1}^{n} \alpha_k \kappa(\beta_k(x-\gamma_k))$

IV. Data Fusion Algorithm

A data fusion algorithm with an adaptive probability of the Kullback information criterion (KIC) is used to fuse the outputs of different models obtained [13]. Kullback information criterion (KIC) is an asymmetric measure for the model selection. Kullback's symmetric or J-divergence is the

sum of two directed divergences, which is the measure of the models dissimilarity and given by Equation (2), [14].

$$KIC(p_i) = \frac{n}{2}\log R_i + \frac{(p_i+1)n}{n-p_i-2} - n\psi\left(\frac{n-p_i}{2}\right) + g(n), \quad (2)$$

where $g(n) = n * \log(n/2)$.

In this research work, we are obtaining multi- linear and nonlinear models using SI for the sEMG sensor for the ring finger motor unit, and two adjacently placed sensors, ring1 and ring2. The probability of each model is computed using the SI model fit value, which is given by $[1 - |Y - \hat{Y}|/|Y - \bar{Y}|] * 100$. The fusion of outputs and adaptive KIC probability is shown in Figure 3 which is followed by the data fusion algorithm.

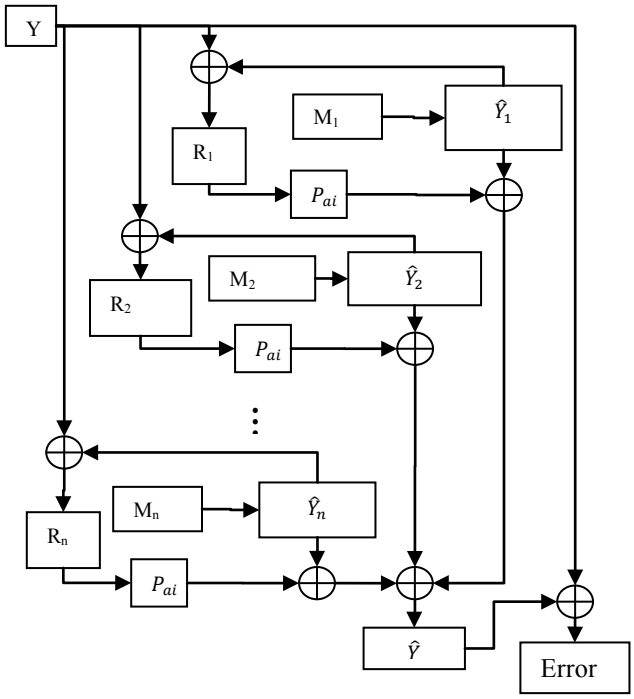


Figure 3. Adaptive KIC Probability Based Data Fusion.

The following fusion algorithm as given by [13] is applied for data fusion of the outputs of different linear and nonlinear models:

1) Identify models $M_1, M_2, \ldots, M_k$ using sEMG data (u) as input and force data (Y) as output, for k number of sensors collecting data simultaneously.
2) Compute the residual square norm

$R_i = \left\|Y - \Phi_i\widehat{\Theta}_i\right\|^2 = \left\|Y - \hat{Y}\right\|$, where $\widehat{\Theta}_i = \{{\Phi_i}^T\Phi_i\}^{-1}{\Phi_i}^T Y$, and

$$\Phi = \begin{bmatrix} Y_p^T & u_p^T & Y_{p-1}^T & \ldots & u_1^T \\ Y_{p+1}^T & u_{p+1}^T & Y_p^T & \ldots & u_2^T \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ Y_{n-1}^T & u_{n-1}^T & Y_{n-2}^T & \ldots & u_{n-p}^T \end{bmatrix}.$$

3) Calculate the model criteria coefficient using Equation (2).
4) Compute the model probability $p(M_i|Z) = \frac{e^{-l_i}}{\sum_{j=1}^{k} e^{-l_j}}$, where l is model selection criterion, i.e. $KIC(p_i)$.
5) Compute the fused model output $\hat{Y}_f = \sum_{i=1}^{k} p(M_i|Z)\hat{Y}_i$.
6) Compute the overall model from $\hat{Y}_f$ and force data.

Here all the computation from step 2) to 6) is adaptive i.e. the residual square norm, $KIC(p_i)$, model probability $p(M_i|Z)$, and fused model output $\hat{Y}_f$ are being updated with time or for each data point.

V. Results and Discussion

An adaptive KIC probability based data fusion algorithm is applied to linear and nonlinear models separately for the models obtained using three sets of input and output data for three sEMG sensors. First the fusion based output for each sensor is obtained and then the three fusion based outputs from three sEMG sensors are fused to get the improved estimates of the skeletal muscle force. The following plots show the improvement in the force estimates in succession with different sensors separately and then combined. All the plots have the measured skeletal muscle force signal in green and the estimated force signal in red color. Figure 4 is a plot of measured and Chebyshev type II filtered finger force data with the adaptive KIC probability fusion based force for the first sensor named as ring1. The two signals are very clear and the estimated signal shows good follow up of the measured signal. Figure 5 shows the measured and Chebyshev type II filtered skeletal muscle force with the adaptive KIC probability fusion based force signals for the second sensor named as ring2. It is evident that the measured signal has a very close follow up by the estimated signal.

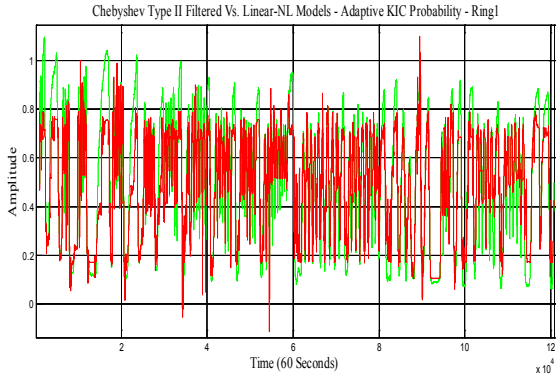


Figure 4. Chebyshev Type II vs. Adaptive KIC Probability Based Force from Linear and Nonlinear Models for Ring1 Sensor.

Figure 6 shows the measured and Chebyshev type II filtered skeletal muscle force with the adaptive KIC probability fusion based force signals for third sensor placed on the motor point of ring finger. The results for this sensor are best among the three based of the used three sensor data and it is evident that the measured signal has a very close follow up by the estimated signal.

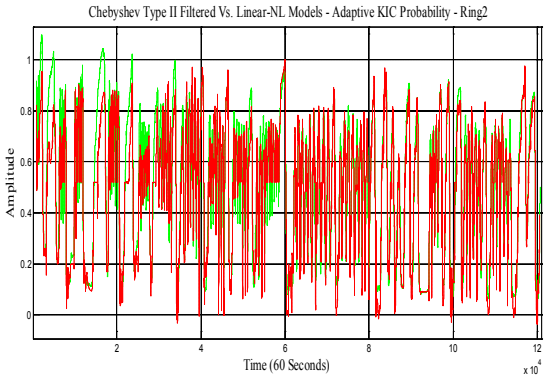


Figure 5. Chebyshev Type II vs. Adaptive KIC Probability Based Force from Linear and Nonlinear Models for Ring2 Sensor.

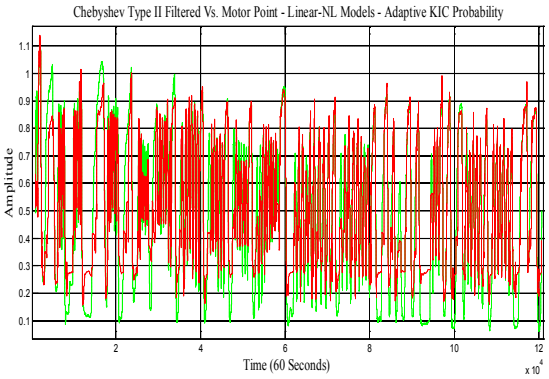


Figure 6. Chebyshev Type II vs. Adaptive KIC Probability Based Force from Linear and Nonlinear Models for Motor Point Sensor.

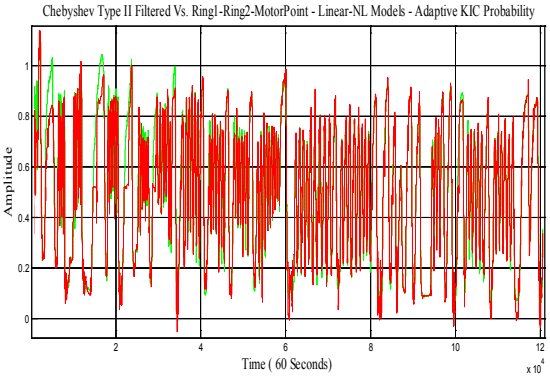


Figure 7. Chebyshev Type II vs. Adaptive KIC Probability Based Force from Linear and Nonlinear Models for Ring1, Ring2 and Motor Point Sensor Combined.

Finally the estimated skeletal muscle force from three sEMG sensors is further fused with adaptive KIC probability based data fusion algorithm. Figure 7 shows the comparison of the measured and Chebyshev type II filter with the final adaptive KIC probability based fusion skeletal muscle force estimate using three sEMG sensors. The final result for three sensors is the best estimate of skeletal muscle force and it is evident that the measured signal has a very close follow up by the estimated signal.

The results show that there is a decrease in the percentage error from the linear and nonlinear model fusion based separate outputs to the fusion based combined output of linear and nonlinear models combined for all three sensors. However, these results show that there is a 16% improvement in the mean fit value of the motor point signal models with the adaptive KIC probability based data fusion algorithm for multi-sensors (three in this case). Figure 8 and 9 show the validation plots of this approach for two separate sets of sEMG and skeletal muscle force data where green signal is the Chebyshev type II filtered measured force signal and the blue signal is estimated using this approach.

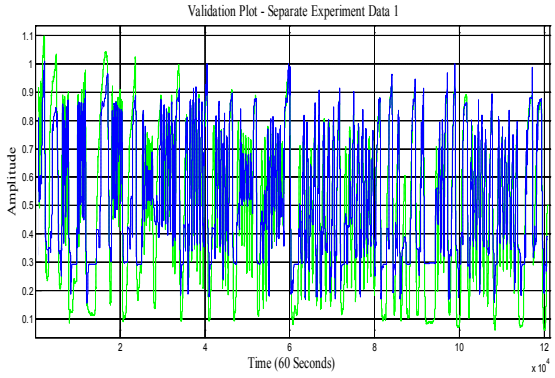


Figure 8. Validation Plot 1 - Chebyshev Type II vs. Estimated Force Signal using Adaptive KIC Probability.

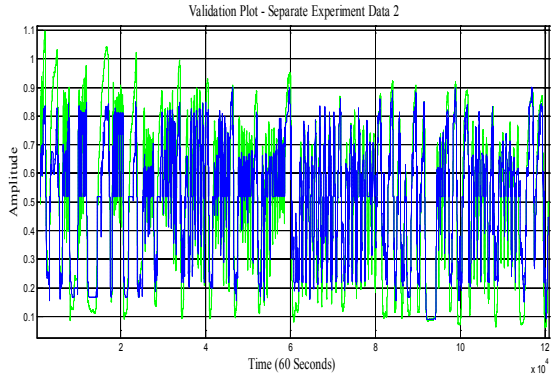


Figure 9. Validation Plot 2 - Chebyshev Type II vs. Estimated Force Signal using Adaptive KIC Probability.

VI. Conclusion and Future Work

Different linear and nonlinear models are obtained using system identification (SI) for three sets of data. Linear and nonlinear models' outputs are fused separately for each sensor and then fused together to get the final force estimate for each sensor. The net force estimates for each sensor is then applied with adaptive KIC probability based data fusion algorithm.

The final estimates of the skeletal muscle force gives improved results. Future work will focus on the improved data collection techniques and capturing data at higher sampling rate (6000 samples per second). Also, we would design new model selection criteria using different model selection criteria such as Akaike Information Criterion (AIC), Kullback Information Criterion (KIC) and the Bayesian Information Criterion (BIC).

ACKNOWLEDGMENT

This research was sponsored by the US Department of the Army, under the award number W81XWH-10-1-0128 awarded and administered by the U.S. Army Medical Research Acquisition Activity, 820 Chandler Street, Fort Detrick MD 21702-5014. The information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred. For purposes of this article, information includes news releases, articles, manuscripts, brochures, advertisements, still and motion pictures, speeches, trade association proceedings, etc.

REFERENCES

[1] Kathryn Ziegler-Graham, PhD, et al., Estimating the Prevalence of Limb Loss in the United States - 2005 to 2050, Archives of Physical Medicine and Rehabilitation, 89 (2008): 422-429.

[2] O'Connor, P., Iraq war vet decides to have second leg amputated, Columbia Missourian, 2009.

[3] N. Dechev, W. L. Cleghorn, and S. Naumann, Multiple finger, passive adaptive grasp prosthetic hand, Mechanism and Machine Theory 36(2001), pp. 1157-1173.

[4] Haruhisa Kawasaki, Tsuneo Komatsu, and Kazunao Uchiyama, Dexterous Anthropomorphic Robot Hand With Distributed Tactile Sensor: Gifu Hand II, IEEE/ASME Transactions on Mechatronics, Vol. 7, No. 3, September 2002, pp. 296-303.

[5] M. Zecca, S. Micera, M. C. Carrozza, and P. Dario, Control of Multifunctional Prosthetic Hands by Processing the Electromyographic Signal, Critical Reviews™ in Biomedical Engineering, 30(4-6), 2002, pp. 459-485.

[6] Claudio Castellini and Patrick van der Smagt, Surface EMG in advanced hand prosthetics, Biological Cybernetics, (2009) 100, pp. 35-47.

[7] P. Kumar, A. Sebastian, C. Potluri, A. Urfer, D. S. Naidu, and M. P. Schoen, "Towards Smart Prosthetic Hand: Adaptive Probability Based Skeletal Muscle Fatigue Model," 32nd Annual International Conference of the IEEE EMBS Conference, 2010.

[8] P. Kumar, C. Potluri, A. Sebastian, S. Chiu, A. Urfer, D. S. Naidu, and M. P. Schoen, "An Adaptive Multi Sensor Data Fusion with Hybrid Nonlinear ARX and Wiener-Hammerstein Models for Skeletal Muscle Force Estimation," 14th WSEAS Int. Conf. on Systems, Vol. 1, pp. 186-191, July 2010.

[9] P. Kumar, C. Potluri, M. Anugolu, A. Sebastian, J. Creelman, A. Urfer, S. Chiu, D. S. Naidu, and M. P. Schoen, "A Hybrid Adaptive Data Fusion with Linear and Nonlinear Models for Skeletal Muscle Force Estimation," 5th Cairo International Conference on Biomedical Engineering, Cairo, Egypt, Dec. 16-18, 2010.

[10] Terence D. Sanger, Bayesian Filtering of Myoelectric Signals, J Neurophysiol, 97, 2007, pp. 1839–1845.

[11] Lennart Ljung, System Identification: Theory for the User. 2nd edition, Printice Hall PTR, 1999, Chap. 1, pp. 1-15.

[12] Lennart Ljung, System Identification ToolboxTM 7 User's Guide, The MathWorks, Inc., 2010.

[13] H. Chen and S. Huang, "A Comparative study on Model Selection and Multiple Model Fusion," 7th International Conference on Information Fusion, pp. 820-826, 2005.

[14] Abd-Krim Seghouane, M. Bekara, G. Fleury, "A Small Sample Model Selection Criterion Based on Kullback's symmetric Divergence," IEEE Transaction, pp. 145-148, 2003.

SESSION

NOVEL APPLICATIONS AND ALGORITHMS

Chair(s)

TBA

Payoff Allocation for PSS Control Service in the Restructured Power System

A. Jalili, H. Shayeghi, H. A. Shayanfar,
Technical Eng. Department
Islamic Azad University- Science and Research Branch, Tehran, Iran

arefjalili@gmail.com, hshayeghi@gmail.com, hashayanfar@yahoo.com,

Abstract- *This paper presents payoff allocation for PSS control service in the restructured power systems. The method is based on cooperative game theory by using the Shapley value concept. Because of the complexity and difficulty associated with the system dynamic performance of the PSS to a financial cost, ITAE performance index is used to evaluate the contribution of the PSS for increasing system stability and it's worth allocation. A three machine power system has been used to illustrate the effectiveness of the proposed technique and the variable payoff shares for the generators under various load conditions. A realistic scenario represented by a daily load curve is also considered to demonstrate how loading conditions affect the payment for the generators. The simulation results show that the proposed methodology provides fair and economically efficient prices for the contribution of each PSS in the stability enhancement of the power systems.*

Keywords: Payoff, PSS, Dynamic Stability, Ancillary Service, Power System.

1 INTRODUCTION

The dynamic behavior of many industrial plants is heavily influenced by disturbances and, in particular, by changes in the operating point. This is typically the case for the restructured power system [1]. Power System Stabilizer (PSS) is a very important ancillary service in power system operation and control for supplying sufficient, stable and reliable electric power with good quality. PSS render a service to the system by providing the damping action when system disturbances influence transmission security and reliability. This damping action is provided through the excitation system of the synchronous generator. In facts, PSS provide a similar service to provide by the governor and excitation systems in maintaining bulk transmission system security. It is therefore reasonable to argue that the PSS should also be considered as an ancillary service provider, within the NERC definition, in exactly the same manner as the generator governor and excitation systems. Such a classification supports the fact that various operation authorities (e.g. WECC) require the installation of the PSS in generators connected to the grid, and at the same time recognizes the importance of the service provided by these controllers. Thus, motivating energy suppliers to properly tune and utilize the PSS to improve, instead of hinder, system security, which is a concern for the system operators now days [2].

Generally, it is not a straightforward exercise to evaluate the worth and contribution of an individual PSS to system welfare due to the complexity and difficulty associated with relating performance of a PSS to a financial cost that would quantify system welfare occurred from a coordinated PSS. Part of the complexity also arises from the way the system performance is measured and valued. To address this issue a game theoretic approach is employed. Game theory has found recent application in power system, particularly in the context of deregulation [3].

There is no doubt that the importance of the PSS to the deregulated system has been clearly understood and appreciated by power system engineers and operators. However, their worth and their contribution enhancement in the system transfer capability and payoffs for the PSS control service have not been investigated and outlined in an analytical manner. Andreoiu et. al [2 , 3] argue the importance of a single PSS in a system is examined in detail and the extent to which it contributes to the system stabilization is made: evidently a coordinated PSS setting will have a beneficial influence on the system performance. In these studies a cooperative game theoretic approach based Shapley values was used to determine the marginal contribution of each PSS in the system. In other words, Shapley values were used to allocate payoffs to each player (i.e. generator equipped with PSS) in the system, depending on how important the PSS is to the overall system stability and security. In Ref. [2] it was shown that the damping factor is used as a cut off value, i.e. after this value the market shall not be cleared; such a criterion may be imposed by the ISO to maintain a reasonable margin of the system security. However, calculation of the damping factor from eigenvalue analysis is very complex due to the fact that the real world power system is large scale system with nonlinear characteristics. Thus, for obtaining the damping factor,

* Correspanding Author. E-Mail Address: hashayanfar@yahoo.com (H. A. Shayanfar)

it is needed to linearzed the model and eigenvalue analysis. In Ref. [3] it was shown that the power system stabilizer control actions be regarded as a system ancillary service toward bulk transmission system security. In this study, the payoff allocation is based on the cooperative game theory, using the concept of Shapley values and the system saving is calculated for each player in the coalitions based on a quadratic performance index, which measures the performance of the PSS for a set of parameters and operating conditions. In this method, the weighting matrix is positive semi-definite and denotes the importance attach to the different state variables in the optimization process. The value of the performance index can be numerically computed when the positive-definite matrix (P) by solving the Lyapunv equation is specified. It should be noted that, solving this equation and obtaining the matrix P is complex and need special process.

Although the cost of the PSS is relatively cheap, its benefits for the system stability should valued to help in convince generating facilities to properly use and tune this controller, since these are no longer under the control of the system operators. Thus, to benefit evaluation and obtain the payoff mechanism for the system stability from the PSS action, it seems that the optimal PSS design is very important. There have been continuing efforts in design of PSS with better performance using various control strategies such as classical [4-5], optimal control method [6], adaptive and variable structure [7-8], robust [9-10] and fuzzy approaches [11-12]. Intelligent and evolutionary based methods [13] are used for the design of the PSS to solve the damping of the power system low frequency oscillations. To overcome this drawbacks and limitation of these methods, such as time consuming and complexity of designing process, Takagi and Sugeno fuzzy Parallel Distributed Compensation (TSFPDC) is used to damping power system oscillations [14]. Here TSFPDC method is accepted and used to evaluate dynamic behavior and financial analysis.

In the present work, we propose the PSS control as a service within the definitions of the system ancillary service and developing appropriate mechanisms for financial compensation to synchronous generators for such services. A cooperative game theoretic approach based on the Shapely values by using ITAE performance index is developed to demonstrate a new payoff allocation for the PSS control service providers. The advantage of this selected performance index is that minimal dynamic plant information is needed, so there is no need for the system model linearization and reducing the complexity of the financial analysis. By the proposed method, the marginal contribution of each PSS and hence how each PSS should be paid for the provided control services will be determined. In other words, Shapley values based on ITAE are used to allocate payoffs to each player in the power system and determine the importance of each PSS in the overall system stability and security.

2 Power system description

The complex nonlinear model related to an n-machine interconnected power system, can be described by a set of differential-algebraic equations by assembling the models for each generator, load, and other devices such as controls in the system, and connecting them appropriately via the network algebraic equations. The structure of the power system component is shown in Fig. 1 to obtain the complete system model for carrying out the transient stability studies. In this system, the excitation power is supplied through a transformer from the generator terminals. High gain exciters may add negative damping to the system and worsen its relative stability. To overcome that, supplementary excitation must be added. In this study, the two-axis model is used for the time domain simulations. Nonlinear model of the multi machine power system is described by set of a equation as given by [15]:

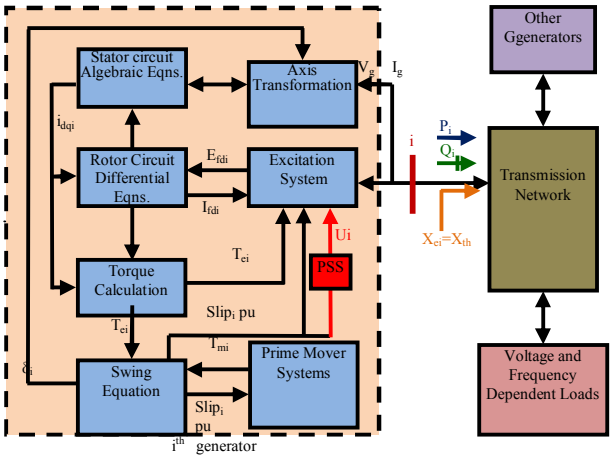


Fig. 1. The structure of the complete power system

$$\dot{\omega}_i = \frac{T_{mi} - T_{ei}}{2H} \tag{1}$$

$$\dot{\delta}_i = \omega_0(\omega_i - 1) \tag{2}$$

$$\dot{E}'_{qi} = \frac{E_{fdi} - (E'_{qi} + (x_{di} - x'_{di})I_{di})}{T'_{doi}} \tag{3}$$

$$\dot{E}_{fdi} = \frac{K_{Ai}(V_{ref} - V_{ti}) + u_{si}) - E_{fdi}}{T_{Ai}} \tag{4}$$

$$T_e = E'_{di}I_{di} + E'_{qi}I_{qi} - (x'_{qi} - x'_{di})I_{di}I_{qi} \tag{5}$$

$$E = E'_{qi}I_{di} - (x_{di} - x'_{di})I_{di} \tag{6}$$

In this study, the three-machine nine-bus power system shown in Fig. 2 is considered as a test system. Details of the system data are given in Ref. [16]. Also, Fig. 3 show the TSFPDC based PSS (see Ref. [14] for more details about the problem solution).

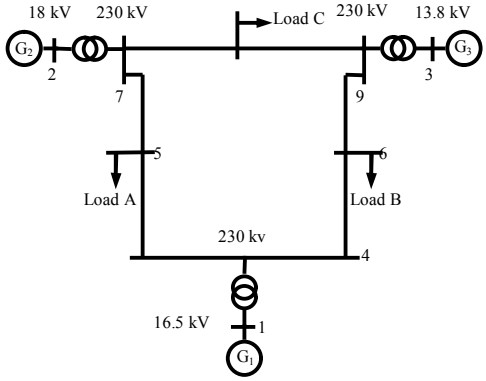


Fig. 2 Three-machine nine-bus power system

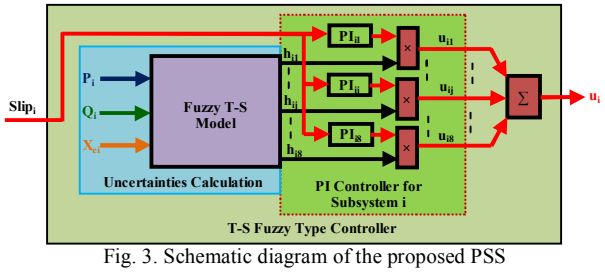


Fig. 3. Schematic diagram of the proposed PSS

In this study, the 3-machine 9 bus power system shown in Fig. 2 is considered as the test system. It is assumed that the set of variables (P,Q,X_e) of certain subsystem varies independently over the following ranges: $P_1 \in [0.45 \quad 1.75]$, $P_2 \in [0.8 \quad 2.35]$, $P_3 \in [0.55 \quad 1.65]$, $Q_1 \in [-0.2 \quad 0.5]$, $Q_2 \in [-0.2 \quad 0.5]$, $Q_3 \in [-0.2 \quad 0.5]$ and $X_{e1} \in [0.4 \quad 0.5]$, $X_{e2} \in [0.4 \quad 0.7]$, $X_{e1} \in [0.4 \quad 0.65]$. The eight local models of proposed method defined according the eight loading condition are given in Table 1. The optimal PI controller gains in the proposed method for the different operation conditions as given in Table 1 are listed in Table 2.

Table 1. The eight loading conditions in per unit

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8
P_{g1}	0.45	0.45	1.75	1.75	0.45	0.45	1.75	1.75
Q_g	-0.2	0.5	-0.2	0.5	-0.2	0.5	-0.2	0.5
X_{eg1}	0.4	0.5	0.4	0.5	0.4	0.5	0.4	0.5
P_{g2}	0.8	2.35	0.8	2.35	0.8	2.35	0.8	2.35
Q_{g2}	-0.2	0.5	-0.2	0.5	-0.2	0.5	-0.2	0.5
X_{eg2}	0.4	0.7	0.4	0.7	0.4	0.7	0.4	0.7
P_{g3}	0.55	1.65	0.55	1.65	0.55	1.65	0.55	1.65
Q_{g3}	-0.2	0.5	-0.2	0.5	-0.2	0.5	-0.2	0.5
$X_{e\,g3}$	0.4	0.65	0.4	0.65	0.4	0.65	0.4	0.65

3. PSS-control ancillary service

In an interconnected and deregulated power system with several generators equipped with PSS, the parameters of the PSS should be optimally tuned in a coordinated manner by the ISO, or a similar entity. The PSSs render a service to the power system by way of providing stabilization action to the small disturbances that occur in the system continually. The stabilization action is done through auxiliary corrective signals to the reference of the automatic voltage regulator. In the absence of this service, the system will most likely become unstable due to the sustained low frequency oscillations.

Table 2. Optimal PI gains for the proposed method

Gain		Gen. 1	Gen. 2	Gen. 3
K_p	Case 1	20.46	20.46	14.22
	Case 2	21.29	21.38	14.31
	Case 3	0.01	20.46	10.28
	Case 4	0.01	21.11	10.33
	Case 5	22.08	22.42	15.15
	Case 6	21.36	23.12	15.66
	Case 7	0.01	22.09	11.04
	Case 8	0.010	22.49	10.56
K_i	Case 1	12.80	2.74	19.80
	Case 2	13.32	2.86	19.90
	Case 3	7.92	19.64	0.18
	Case 4	8.28	20.26	0.18
	Case 5	13.818	3.08	21.09
	Case 6	13.36	3.10	21.77
	Case 7	8.4578	21.12	0.19
	Case 8	8.6046	21.58	0.18

PSS render a service to the system by providing damping action when system disturbances influence transmission security and reliability. This damping action is provided through the excitation system of the synchronous generator. In facts, PSS controllers provide a similar service to that provided by the governor and excitation systems in maintaining bulk transmission system security. It is therefore reasonable to argue that the PSS should also be considered an ancillary service provider, within the NERC definition, in exactly the same manner as the generator governor and excitation systems. Such a classification supports the fact that various operation authorities (e.g. WECC) require the installation of the PSS in generators connected to the grid, and at the same time recognizes the importance of the service provided by these controllers. Thus, motivating energy suppliers to properly tune and utilize the PSS to improve, instead of hinder, system security, which is a concern for system operators now a days [2].

3.1. Payoff allocation and pricing of PSS-control service

It is claimed that the increasing competition in the power market can help to maximize customer's payoffs. This claim is conceivably supported by the results obtained from the application of game theory. Although, game theory has been associated with parlor games, most research in game theory has focused on how groups of people interact. The game can be classified into two major categories, namely cooperative or a noncooperative game. A noncooperative game takes place when any coalition is interested in maximizing its own payoffs regardless of the payoffs for the counter coalition. Within a coalition, strategies are coordinated among participants in a cooperative game. In other words, in noncooperative games, participants make no commitments to coordinate their strategies. Conversely, in cooperative games, participants can make credible commitments to coordinate their strategies. In a

noncooperative game, some participants choose a strategy while other participants try to identify their best response to that strategy. In a cooperative game, participants define strategies that would lead to the best outcome for the coalition [17].

In the deregulated power systems, it would be the responsibility of the ISO or a similar entity to evolve a coordinated PSS tuning and operation strategy based on certain system wide objective function. Therefore, the PSSs operation can be modeled as a cooperative game, having as its characteristic function a savings formulation, which is based on the objective function used in the tuning process, and where one or more generators are considered together (in all possible coalitions) to obtain the fair revenue allocation. Thus, our analysis of the worth of the PSS control ancillary service should be based on two important issues: (a) how the coalitions are formed amongst the PSS and consequently (b) how the benefit from a PSS service is allocated. In these two interrelated issues, our main concern is to obtain the most likely outcome from various game situations. Especially, when it comes to the distribution of the savings due to the PSS-control action, the revenue corresponding to a PSS in a particular coalition is very difficult to evaluate. An intuitively attractive solution concept for the *n*-person cooperative games with transferable utility (payoff, in our case) has been proposed by Shapley in 1953 and is called the Shapley value. The advantage in the use of Shapley value is that it represents a direct and simple manner to find a solution for a cooperative game. The marginal contribution Ψ of a player i in coalition C ($\forall\ i \in C$) will be given by [3]:

$$\Psi_i(C)=v(C)-v(C/\{i\}) \quad (7)$$

Where, $v(C)$ is the payoff resulted from the coalition C. The Shapley value φ, which is the weighted average of the marginal contributions of a player i in all possible coalitions, is hence given by:

$$\varphi_i = \sum_C C_w(C)\psi_i(C) \quad (8)$$

Where, $C_w(C) = \dfrac{(q-1)!(n-q)!}{n!}$, q is the size of a coalition.

Having obtained the contribution of a PSS to system savings, a mechanism for financial compensation to the synchronous generators for their PSS-control service can now be devised for the ISO. The payment ρ we propose here has the following structure:

$$\rho_i=\rho_{ai}+\varphi_i \quad (9)$$

In (9), ρ_{ai} represents the component of payment associated with availability of the PSS at a generator. This component is payable to the generator for having the PSS installed and adhering to the ISO's instructions on the parameter settings. The generator is entitled to this component of payment even if the ISO instructs that the said PSS remains off-line. The second component of ρ given in (9) denotes the variable payment component as given in (8), proportional to the "worth" of the PSS in system stabilization, and is determined using the proposed Shapley value-based method.

Since, PSS action can be also technically classified as an ancillary service. However, the dynamic services provided by this controller are an issue, because one of the key features of this controller is its dynamic response characteristics. Thus, the PSS controller should be first optimally tuned to optimize the market operating conditions, *i.e.* power dispatch, price levels and adequate system security. Although, the cost of the PSS is relatively cheap, its benefits for system stability should valued to help convince generating facilities to properly use and tune this controller, since these are no longer under the control of the system operators. Thus, to benefit evaluation and obtain payoff mechanism for the system stability from the PSS action, it seems that the optimal PSS design is very important.

4. System payoff analysis

4.1 System transfer capability and payoffs for PSS control service

With all PSS online, the system loading is now increased gradually and uniformly at all buses using a Load Factor (LF) that denotes the load increase with respect to the base load. This load increase scenario is used only to illustrate the effect of the PSS on the system transfer capability and improve power system performance. The dominate ITAE performance index of the system shift towards instability as the system loading is increased, or in other words, the ITAE increase monotonically as LF increases. Fig. 4 demonstrates ITAE performance index value based on LF for each feasible coalition (C_2, C_{12}, C_{23} and C_{123} are feasible coalition). A typical ITAE (ITAE=2) is considered as a cut off value, i.e. beyond which the market shall not be cleared; such a criterion may be imposed by the ISO to maintain a reasonable margin of system stability and security. This is demonstrated in Fig. 4.

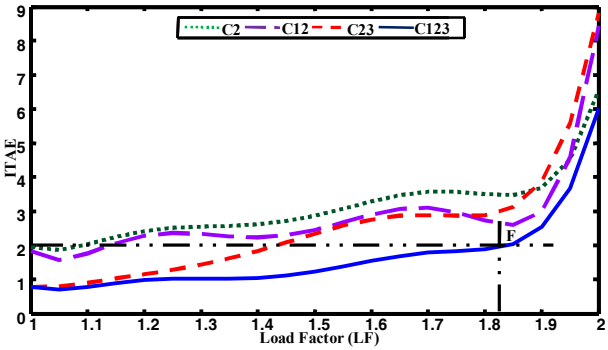


Fig. 4. ITAE- LF curve for normal operation condition

To determine the payoff allocation that described in sec. 3, Fig. 5 shows the flowchart of pricing the PSS

control service. By setting the PSSs to offline in all possible combinations, all the feasible coalitions (PSSs manage to maintain the system stability) in which the three PSS may operate are obtained. Coalition C-123 denoted the grand coalition; the maximum LF up to which the market may be cleared if the load would increase uniformly (the point F where ITAE intersects the cut off value ITAE=2) is 1.8015 pu, which corresponds to a total of 567.48 MW (system base loading is 315 MW), that in turn translates to an improvement in the system transfer capability of 225.48 MW with respect to base loading.

Observe from Fig. 3 that, as expected, different coalitions have different effect on the system transfer capability. Consequently one can also examine their effect on system transfer capability, as explained here for coalition C-123. Thus, assuming that the worth of unit increase in system transfer capability with respect to the base loading is 10 \$/MW, i.e. δ_{MW} is10 \$/MW, the benefit from specific PSS coalition δ_c can be calculated. The value of δ_{MW}=10 \$/MW used here is only to demonstrate the proposed method; in real system, the ISO would be required to determine/negotiate the actual value δ_{MW}, i.e. the worth given is only being used as a relative measure of how secure the system is.

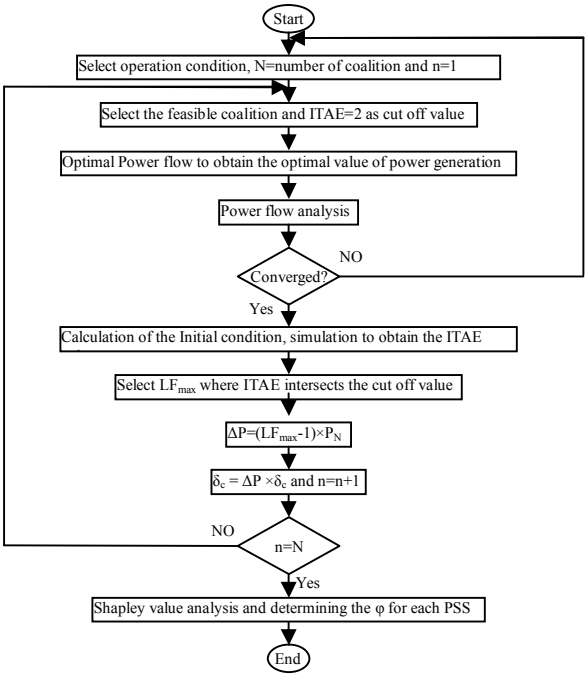


Fig. 5. Pricing the PSS control service to payoff allocation

As the market is cleared at this loading condition, the individual generators would be entitled to an additional payment in return for providing the PSS control ancillary service and improving the system transfer capability. Consequently, the ISO's problem is to allocate the total worth δ_c achieved from the PSS operation in a fair and rational manner. Table 3, based on each coalition's benefits δ_c outlines the application of the Shapley value criterion introduced earlier and yield the fair share of payment each PSS would be entitled to, in dollars, for a 315 MW loading level. Note that, for each PSS the value of φ_{PSS} denote the terms of the Shapley value (8); δ_c=0 for infeasible coalition, and dashes are used to identify for each PSS the coalitions in which it does not participate (e.g. PSS-1 does not take part in coalition C-1, C-3 and C-13). Also, this analysis would depend on the actual market clearing conditions, i.e. actual generation and load levels, which change through the day.

Table. 3 Shapley value based payment allocation to generators

	C-1	C-2	C-3	C-12	C-13	C-23	C-123	Shapley Value
δ_c	------	181.6	------	423.9	------	1417.6	2524.8	
C_w	1/3	1/3	1/3	1/6	1/6	1/6	1/3	
PSS1								
Ψ	------	0	------	242.3	------	0	1107.2	409.45
Φ_{pss1}	------	0	------	40.38	------	0	369.07	16.2%
PSS2								
Ψ	------	181.6	------	423.9	------	1417.6	2524.8	1209.05
Φ_{pss2}	------	60.53	------	70.65	------	236.27	841.6	47.89%
PSS3								
Ψ	------	0	------	0	------	1236	2100.9	906.3
Φ_{pss3}	------	0	------	0	0	206	700.3	35.91%
Total Payment, ρ								2524.8

From the Table 3 it can be seen that the Shapley values, and consequently the variable payment components for generator 1, 2 and 3 are 409.45\$, 1209.05\$ and 906.3\$, respectively. This translates in to a 16.2% share of the payoff to generator 1, 47.89% to generator 2 and 35.91% to generator 3 in the variable component. The positive small variable component of payment for introducing an overall detrimental effect on the system transfer capability. This is evident from the fact that PSS1 has a small marginal contribution in coalition C12 and C123.

4.2 System transfer capability and payoffs considering contingencies

Once a method of determining the payoffs to participating generators is in place it would be of interest to investigate if such a payoff obtained for the normal operating condition at a given time will still be fair allocation in a contingency state when there is a change in system topology. This is important, since typically an N-1 contingency criterion is used to define system security levels [2]. To this effect set of contingencies represented by outage of one transmission line at a time was considered. Note that the outage of line 4-5 was not included since there was no feasible power flow solution for this contingency, which would lead to a voltage collapse problem that is beyond the scope of this paper. Hence, the following contingencies are considered: 4-6, 5-7, 6-9, 7-8 and 8-9 denoted by CTG1 – CTG5, respectively. Table 4 shows, for all

feasible coalitions and considering all contingencies, the amount of power, over the above the nominal load that the system can serve without infringing any operation constraints, while ITAE<2 at the previous 315MW loading level. The proposed financial mechanism is used to determine the worth of each PSS in various operating condition and system configurations.

Table 5. Amount of power in all feasible coalitions by considering all contingencies

	CTG1	CTG2	CTG3	CTG4	CTG5
C2	0	-55.4	0	0	0
C12	0	-16.46	0	0	0
C23	232.4	194.3	194.46	180	272.46
C123	223.7	193.85	193.21	184.49	273.33

Fig. 6 shows the variable payoff shares for the generators that provided PSS control service for the nominal operation condition and under contingencies. It can be seen that how the system topology affects the payment allocation.

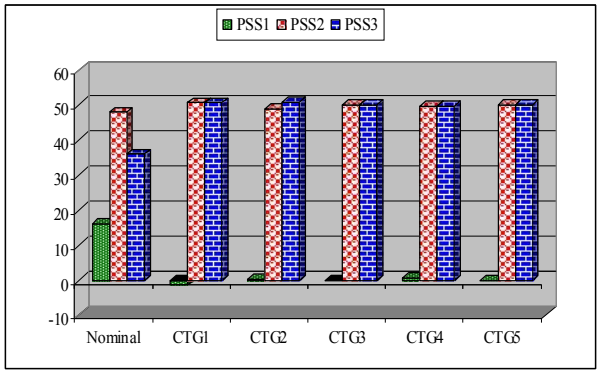


Fig. 6. Variable payoff shares for the generators in all condition

5. Conclusions

This paper presents worth allocation of PSS and its contribution enhancement in system transfer capability and new payoff mechanism for the PSS control service in the deregulated environment. The payment scheme is based on the cooperative game theory by using the Shapley value. In this method, the ITAE performance index is used to associate the system dynamic behavior and cost of the PSS. The effectiveness of the proposed technique was tested on the 3 machine 9 bus power system and variable payoff shares for the generators in return from the PSS control service for the nominal operating condition and under contingencies are obtained. By considering the daily load curve, it can be seen that the loading conditions affect the payment. The simulation results show that the proposed methodology provides fair and economically efficient prices method.

References

[1] Raineri, S. Rios and D. Schiele, "Technical and economic aspects of ancillary services markets in the electric power industry: An interconnected comparison," Energy Policy, Vol. 34, pp. 1540-1555, 2006.

[2] A. Andreoiu, K. Bhattacharya and C. Canizaress, "Pricing power system stabilizers using game theory," IEE Proceeding Generation Transmision Distribution, Vol. 152, No. 6, pp. 780-786, 2005.

[3] A. Andreoiu and K. Bhattacharya, "PSS control as an ancillary service," Electric Power System Research, Vol. 74, pp. 391-399, 2005.

[4] U. Yu, "Electric power system dynamics" Academic Press, 1983.

[5] Y Y. Hsu C. Hsu, "Design of proportional integral power system stabilizer," IEEE Trans Transaction on Power Systems, Vol. 1, No. 2, pp. 46-53, 1986.

[6] E. Motamedi, P. Naderi, T. Mohammadi and M. Norman, "Kalman filter application for linear optimal control and dynamical stability of multi machine power system using optimal placement of stabilizers," International Review of Electric Engineering, Vol. 4, No. 4, pp. 570-576, 2009.

[7] D. Pierre, "A perspective on adaptive control of power systems," IEEE Transaction Power System PWRS, Vol. 2, No. 2, pp. 387-396, 1987.

[8] S. Cheng, O. Malik and G. Hope, "Damping of multi-modal oscillations in power systems using a dual-rate adaptive stabilizer" IEEE Transaction on Power System, PWRS, Vol. 3, pp. 101-108, 1988.

[9] A. Hardiansyah, S. Furuya and J. Irisaw, "A robust H∞ power system stabilizer design using reduced-order models," Electrical Power Energy System, Vol. 28, pp. 21-28, 2006.

[10] A. Ghany, "Design of a mixed H2/H∞ robust PID power system stabilizer with fuzzy adaptation and simulated annealing optimization," IEEE Transaction on Energy Conversion, pp. 316-324, 2008.

[11] D. Chatuved and O. Malik, "Neuro fuzzy power system stabilizer," IEEE Transaction on Energy Conversion, Vol. 23, No. 3,pp. 887-894, 2008.

[12] A. El-zonkoly, A. Kalil and N. Ahmied, "Optimal tuning of lead lag fuzzy logic power system stabilizer using particle swarm optimization," Expert System with applications, Vol. 10, pp1-10, 2008.

[13] S. Akbari and K. Amooshahi "Power system stabilizer design using evolutionary algorithms," International Review of Electric Engineering, Vol. 4, No. 5, pp. 925-931, 2009.

[14] A. Jalili, H. Shayeghi and H. A. Shayanfar, "T-S fuzzy parallel distribution compensation controller for power system stabilizer," 5th International Conference on Technical and Physical Problems of Power Engineering Spain, pp. 180-174, 2009.

[15] K. Padyar, "Power system dynamics stability and control," BS Publication, Second Edition 2006.

[16] P. Anderson and A. Fouad, "Power system control and stability," Ames, IA: Iowa State Univ. Press, 1977.

[17] M. Shahidehpour, H. Yamin and Z. Li, "Market operation in electrical power system," John wiley & Sons Inc Publication, 2002.

Biographies

Aref Jalili received the B.S. and M.S.E Degrees in Electrical Engineering from Azad University, Ardabil and South Tehran branches, Iran in 2003 and 2005, respectively. Currently, he is a Ph. D. student in Electrical Eng. Department of the Azad University. Science and Technology Research Branch, Tehran, Iran. His Areas of Interest in Research are the Application of Artificial Intelligence Techniques to Power System Control and operation.

Hossein Shayeghi Received the B.S. and M.S.E. Degrees in Electrical and Control Engineering in 1996 and 1998, respectively. He received his Ph. D. Degree in Electrical Engineering from Iran University of Science and Technology, Tehran, Iran in 2006. Currently, he is an Associate Professor in Technical Eng. Department of University of Mohaghegh Ardabili, Ardabil, Iran. His research interests are in the Application of Robust Control, Artificial Intelligence to Power System Control Design, Operation and Planning and Power System Restructuring. He has authored five books in Electrical Engineering area, one in Power Systems Analysis, one in MATALAB, one in Electric Circuits, one in Electric DC Machines, one in Electric Installations (all in Persian language). Also, he is co-authored of a book chapter- "A Review on Load Frequency Control Strategies" in the book "Complex Behaviour of the Distributed Generation System**:** Intelligent Management of the Renewable Energy Resources for assuring the DG System Power Quality and a Sustainable Development". He has published more than 160 technical papers in International Journals and Conferences Proceedings. He is a member of Iranian Association of Electrical and Electronic Engineers (IAEEE) and IEEE. He is a member of Iranian Association of Electrical and Electronic Engineers and IEEE.

Heidarali Shayanfar Received the B.S. and M.S.E. Degrees in Electrical Engineering in 1973 and 1979, respectively. He received his Ph. D. Degree in Electrical Engineering from Michigan State University, U.S.A., in 1981. Currently, He is a Full Professor in Electrical Engineering Department of Iran University of Science and Technology, Tehran, Iran. His Research Interests are in the Area of Application of Artificial Intelligence to Power System Control Design, Dynamic Load Modeling, Power System Observability Studies, Voltage Collapse, Congestion Management in a Restructured Power System, Reliability Improvement in Distribution Systems and Reactive Pricing in Deregulated Power Systems. He has published more than 350 technical papers in International Journals and Conferences Proceedings. He is a Member of the Iranian Association of Electrical and Electronic Engineers and IEEE.

PSO-TVAC Algorithm for Multi Objective PSS Design in Multi-Machine Power System

A. Ghasemi
Technical Eng. Department, University of Mohaghegh Ardabili, Ardabil, Iran
E-mail: ghasemi.agm@gmail.com

O. Abedinia
Islamic Azad University, Young Researcher Club, Ardabil Branch, Ardabil, Iran
E-mail: oveis.abedinia@hotmail.com

H. A. Shayanfar *
Electrical Eng. Department, Islamic Azad University, South Tehran Branch, Tehran, Iran
E-mail: hashayanfar@yahoo.com

Mohammad S. Naderi
School of Electrical Eng. & Telecommunication, UNSW, Sydney, Australia
E-mail: m.salaynaderi@unsw.edu.au

Abstract— **Particle Swarm Optimization with Time Varying Acceleration Coefficients (PSO-TVAC) is used to optimize Power System Stabilizer (PSS) parameters for a multi objective power system in this paper. In the proposed Syndicate tuning technique, two performances indicates as ITAE and FD are computed for the stability and performance at each of the given set of operating conditions of the system simultaneously. This approach facilitates easy handling of the multiple system models. Thereby, yielding robust and reliable stabilizer parameters. Since the objectives are not the same, a PSO-TVAC technique is used to calculate the best solution. The plausibility of the proposed algorithm is demonstrated and its performance is compared with other methods on a 3 machine 9 bus standard power system. Simulation results illustrate that the proposed algorithm has better outperforms the other algorithms.**

Keywords: PSS, PSO-TVAC, Multi Objective, Multi-Machine Systems.

I. INTRODUCTION

The dynamic stability of power systems is an important factor for secure system operation. The action of a Power System Stabilizer (PSS) is to extend the angular stability limits of a power system by providing supplemental damping to the oscillation of the synchronous machine rotors through the generator excitation. This damping is provided by an electric torque applied to the rotor that is in phase with the speed variation. Power system instabilities can arise in certain circumstances due to the negative damping effects of the PSS on the rotor, which is based on tuning PSSs around a steady-state operating point; their damping effect is only valid for the small excursions around this operating point. During severe disturbances, a PSS may actually cause the generator under its control to loose synchronism in an attempt to control its excitation field [1].

The modification of low frequency fluctuation by the application of PSS has been widely studied. Principled and systematic tuning procedures for the conventional lead compensation PSS have been established [2] and successfully applied by the utilities.

Currently, many plants prefer to employ conventional lead-lag structure PSSs, due to the ease of online tuning and reliability [3]. However, these controllers don't have appropriate revenue in different load conditions. Consequently, a lot of intelligent methods have been introduced to optimal tuning of the PSSs parameters [4].

Ant Colony (AC) is a powerful optimization technique which is based on the behavior of the artificial ants is inspired from real ants [5]. This algorithm works concurrently and independently and collective interaction via indirect communication leads to good solutions. The greedy heuristic helps find acceptable solution in the early solution in the early stages of the search process [6]. However, this algorithm has some disadvantage points as slower convergence than other heuristics and no centralized processor to guide the AS towards good solutions.

Moreover Artificial Neural Network (ANN) is another intelligent method which is used for the PSS tuning. Two reasons are put forward for using ANN. First, since an ANN is based on parallel processing, it can provide extremely fast processing facility. The second reason for the high level of interest is the ability of ANN to realize complicated nonlinear mapping from the input space to the output space [7]. This technique has its own advantages and disadvantages. According to this fact that, the performance of power system is improved by ANN based controller but, the main problem of these controllers are the long training time and selecting the

* Correspanding Author. E-Mail Address: hashayanfar@yahoo.com (H. A. Shayanfar)

number of layers and number of neurons in each layers [8].

Hence, in this paper to overcome these problems, a Particle Swarm Optimization with Time Varying Acceleration Coefficients (PSO-TVAC) is used for the solution of the PSS problem. Here, the PSO-TVAC optimization algorithm is used for the optimal tuning of the PSS parameters according to the multi objective functions to improve the optimization synthesis and damping low frequency oscillations following disturbances in the power systems. In many PSS optimization problems there are several objectives to optimize. For such multi-objective problems, there is not usually a single best solution but a set of solutions that are superior to others when considering all objectives. This multiplicity of the solutions is explained by the fact that the objectives are generally conflicting ones [9]. Therefore, this paper is considered the multi objective functions as a Time multiplied Absolute value of the Error (ITAE) and Figure of Demerit (FD) to find the optimum parameters of the PSS using the advantages of the PSO-TVAC.

The effectiveness of the proposed PSO-TVAC PSS is tested on a 3-9 bus multi-machine power system under different operating conditions which is compared with the CPSS [10] and PSO PSS (TSPSS) [11] through nonlinear time simulation and some performance indices. The simulation results show that the proposed technique for optimizing parameters can improve the dynamic stability of the power system and effectively damped the low frequency oscillation under different load conditions.

II. PROBLEM STATEMENT

A. Three Machine Nine Buses Power System

In this study, the three-machine nine-bus power system shown in Fig.1 is considered. Details of the system data are given in Ref. [11]. Furthermore, operation of the system is tested in three different conditions as nominal, lightly and heavily loading conditions.

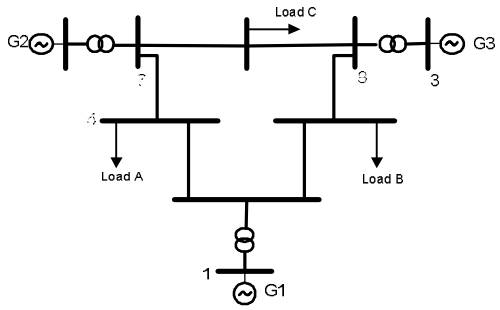


Figure 1. Three-machine nine-bus power system.

B. CPSO Technique

Classic PSO (CPSO) is one of the optimization techniques and a kind of evolutionary computation technique which is launched by the Aberhart Rasel. The method has been found to be robust in solving problems featuring nonlinearity and non-differentiability, multiple optima, and high dimensionality through adaptation, which is derived from the social-psychological theory. The features of the method are as follows [12]:

- The method is developed from research on swarm such as fish schooling and bird flocking.
- It is based on a simple concept. Therefore, the computation time is short and requires few memories [11].
- It was originally developed for nonlinear optimization problems with continuous variables. It is easily expanded to treat a problem with discrete variables.

CPSO is basically developed through simulation of bird flocking in two-dimension space. The position of each agent is represented by XY axis position and also the velocity is expressed by VX (the velocity of X axis) and VY (the velocity of Y axis). Modification of the agent position is realized by the position and velocity information [13]. Bird flocking optimizes a certain objective function. Each agent knows its best value so far (pbest) and its XY position. This information is analogy of personal experiences of each agent. Moreover, each agent knows the best value so far in the group (gbest) among pbest. This information is analogy of knowledge of how the other agents around them have performed. Namely, each agent tries to modify its position using the following information [13]:

- The current positions (x, y),
- The current velocities (VX, VY),
- The distance between the current position and pbest
- The distance between the current position and gbest

This modification can be represented by the concept of velocity and the place of that. Velocity of each agent can be modified by the following equation:

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (1)$$

$$V_i(t+1) = \omega v_i(t) + c_1 r_1(t)[pbest_i(t) - x_i(t)] + c_2 r_2(t)[leader_i(t) - x_i(t)] \quad (2)$$

Where,

- x_i: position of agent i at iteration k
- v_i: velocity of agent i at iteration k

- w: inertia weighting
- $c_{1,2}$: tilt coefficient
- $r_{1,2}$: rand random number between 0 and 1
- leader: archive of unconquerable particles
- pbesti: pbest of agent i
- gbest: gbest of the group

Convergence of the PSO strongly depended on w, c1 and c2. While c1,2 are between 1.5 till 2, however the best choice to these factors is 2.05. Also, $0 \le w < 1$ whereas this value is really important factor to the system convergence and this is better that this factor define dynamically. While it should be between 0.2 and 0.9 and it should decrease linear through evolution process of population. Being extra value of w at first, provides appropriate answers and small value of that help the algorithm to convergence at the end [12].

C. PSO with Time-Varying Inertia Weight (PSO-TVIW)

The PSO-TVIW method is capable of locating a good solution at a significantly faster rate, when compared with other evolutionary optimization methods; its ability to fine tune the optimum solution is comparatively weak, mainly due to the lack of diversity at the end of the search [14]. Also, in PSO, problem-based tuning of parameters is a key factor to find the optimum solution accurately and efficiently. The main concept of PSO-TVIW is similar to CPSO in which the Eqs. (1), (2) are used. However, for PSO-TVIW the velocity update equation is modified by the constriction factor C and the inertia weight w is linearly decreasing as iteration grows [15].

$$V_i(t+1) = C\{\omega v_i(t) + c_1 r_1(t)[pbest_i(t) - x_i(t)] + c_2 r_2(t)[leader_i(t) - x_i(t)]\} \quad (3)$$

$$\omega = (\omega_{\max} - \omega_{\min}).\frac{(k_{\max} - k)}{k_{\max}} + \omega_{\min} \quad (4)$$

$$C = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|}, \text{where } 4.1 \le \varphi \le 4.2$$

D. PSO with Time-Varying Acceleration Coefficients (PSO-TVAC)

Consequently, PSO-TVAC is extended from the PSO-TVIW. All coefficients including inertia weight and acceleration coefficients are varied with iterations. The equation of PSO-TVAC for velocity updating can be expressed as [16]:

$$V_i(t+1) = C\{\omega v_i(t) + ((c_{1f} - c_{1i})\frac{k}{k_{\max}} + c_{1i}).r_1(t)[pbest_i(t) - x_i(t)] + ((c_{2f} - c_{2i})\frac{k}{k_{\max}} + c_{2i}).r_2(t)[leader_i(t) - x_i(t)]\}$$

The procedure of PSO-TVAC for tuning PSS parameters is described in fig.2.

E. Applying PSO-TVAC Algorithm to Power System

For simplicity, a conventional PSS is modeled by two identical stages, lead/lag network which is represented by a gain K_s and four time constants T_1, T_2, T_3 and T_4 [3]. Fig. 3 shows the block diagram of the PSS. In this study the proposed controller is connected to G1, G2 and G3 while $T_{2,4}$ are fix.

The simulation operated with multi objective with PSO-TVAC algorithm and the objective functions for optimization as follow:

$$f_1 = \sum_{j=1}^{N_l}\sum_{i=1}^{N_g} (200 \times OS_{ij})^2 + (500 \times US_{ij})^2 + 0.08 \times T_{s,ij}^2$$

$$f_2 = 100 \times \sum_{j=1}^{N_l}\sum_{i=1}^{N_g} \int_0^{t_{sim}} t(|\Delta\omega_{ij}|).dt$$

In this study the FD and ITAE are presented by f1 and f2, respectively. Accordingly, Fig. 4 shows the trend of objective function's variation. Also, the parameters for PSO-TVAC algorithm are presented in Table 2.

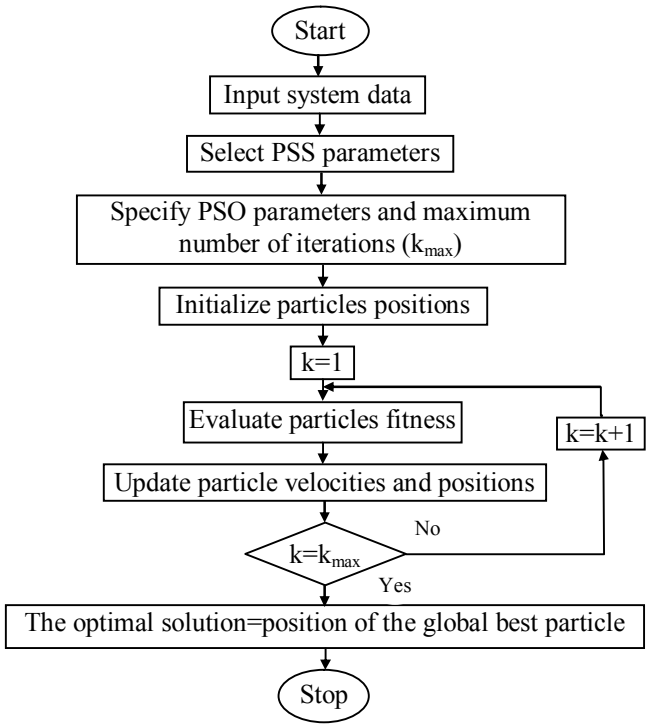


Figure 2. Flowchart of the PSO-TVAC

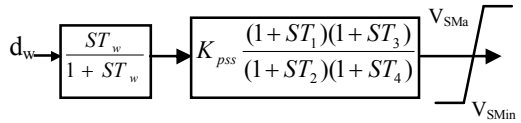


Figure 3. The Block Diagram of PSS

TABLE I. PARAMETERS FOR PSO-TVAC ALGORITHM

C_{1f}	0.2
C_{1i}	2.5
C_{2f}	2.5
C_{2i}	0.2
φ	4.1
w_{min}	0.4
w_{max}	0.9
Population	40
Iteration	100

Table 2 shows the low and up boundaries of the parameters. Results of the PSS parameter set values for the proposed fitness functions are given in Table 3. Moreover, the information of operating conditions is presented in Table 4.

TABLE II. DOMAIN PARAMETERS PSS FOR OPTIMIZATION

	K_{pss}	T_1	T_2	T_3	T_4
Min value	0.1	0.01	0.01	0.01	0.01
Max value	50	5	5	5	5

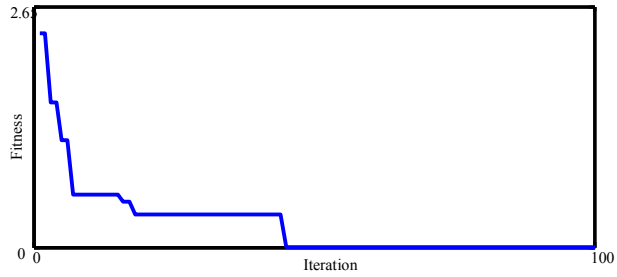


Figure 4. Objective function variation of PSSs design

TABLE III. THE RESULT OF OPTIMIZED PARAMETERS

method	Gen	Kpss	T1	T2	T3	T4
CPSS	G1	18	0.09189	0.05	0.09189	0.05
	G2	16	0.10191	0.05	0.10191	0.05
	G3	20	0.08615	0.05	0.08615	0.05
PSO	G1	46.21	0.06	0.05	0.2383	0.05
	G2	36.87	0.1097	0.05	0.185	0.05
	G3	16.66	0.2257	0.05	0.071	0.05
PSOTVAC	G1	39.631	0.310	0.011	0.41	0.033
	G2	30.911	0.222	0.012	0.34	0.054
	G3	30.931	0.229	0.015	0.24	0.034

TABLE IV. GENERATOR OPERATING CONDITIONS (IN PU)

Gen	Nominal		Heavy		Light		Vt(pu)
	P(pu)	Q(pu)	P(pu)	Q(pu)	P(pu)	Q(pu)	
G_1	0.72	0.27	2.21	1.09	0.36	0.16	1.040
G_2	1.63	0.07	1.92	0.56	0.80	-0.11	1.025
G_3	0.85	-0.11	1.28	0.36	0.45	-0.20	1.025

III. NONLINEAR TIME-DOMAIN SIMULATION

To assess the effectiveness and robustness of the proposed controller, simulation studies are carried out for the various fault disturbances and fault clearing sequences for two scenarios. The numerical results of performance robustness for all cases are listed in Table 5-6. It can be seen that the values of this system performance characteristics with the proposed controller are smaller compared to that other compared methods.

A. Scenario 1

In this scenario, performance of the proposed controller under transient conditions is verified by applying a 3-cycle three-phase fault at t=1 sec, on bus 7 at the end of line 5-7. The fault is cleared by permanent tripping the faulted line. Speed deviations of the generators G1, G2 and G3 under light load condition are shown in Fig. 5.

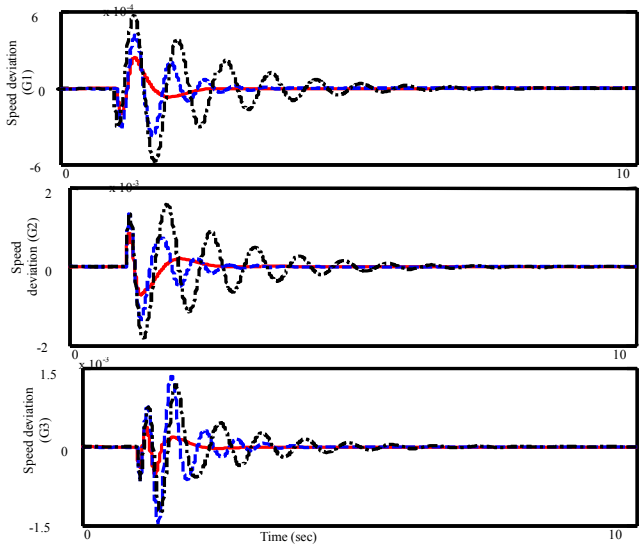


Figure 5. System response under light loading condition in scenario I; Solid (PSO-TVAC), Dashed (PSO), Dotted (CPSS)

B. Scenario 2

In this scenario, another severe disturbance is considered for different loading conditions; while, a 3-cycle, three-phase fault is applied at the same location in scenario I. The fault is cleared without line tripping and the original system is restored upon the clearance of the fault. Fig. 6 shows the result of simulation in nominal load condition.

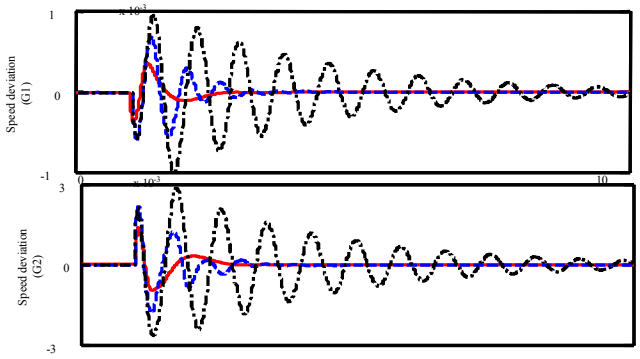


Figure 6. System response under nominal loading in scenario II; Solid (PSO-TVAC), Dashed (PSO), Dotted (CPSS)

TABLE V. VALUE OF ITAE IN DIFFERENT TECHNIQUES

	Scenario 1			Scenario 2			Scenario 3		
Method	Nominal	Light	Heavy	Nominal	Light	Heavy	Nominal	Light	Heavy
PSOTVAC	0.3202	0.1821	0.3055	0.3111	0.1685	0.3003	0.3167	0.4235	0.1977
PSO	0.5319	0.3060	0.5219	0.5088	0.2862	0.4884	2.1373	0.6176	2.7168
CLASSIC	4.2219	0.9664	5.5183	4.1444	0.9278	5.4732	2.3371	0.7331	3.4248

TABLE VI. VALUE OF FD IN DIFFERENT TECHNIQUES

	Scenario 1			Scenario 2			Scenario 3		
Method	Nominal	Light	Heavy	Nominal	Light	Heavy	Nominal	Light	Heavy
PSOTVAC	0.7288	0.4969	0.5005	0.6981	0.4700	0.4781	0.4466	0.6583	0.4297
PSO	1.8419	1.2479	1.9159	1.7849	1.1933	1.8039	2.7363	0.8125	4.1087
CLASSIC	2.5105	2.3375	3.1590	2.4450	2.0510	3.0541	3.0639	1.1805	6.1082

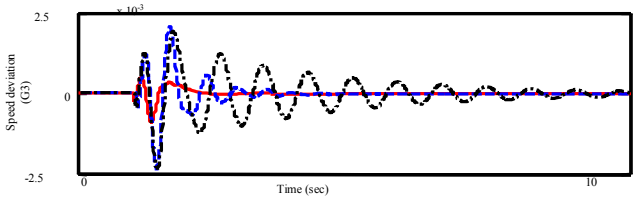


Figure 6. System response under nominal loading in scenario II; Solid (PSO-TVAC), Dashed (PSO), Dotted (CPSS)

C. *Scenario* 3

It is very important to test the PSS under the loading power factor operating condition. A 0.2 p.u. step increase in the mechanical torque was applied at t=1.0. Fig. 7 shows the result of simulation that is tested in heavy load condition.

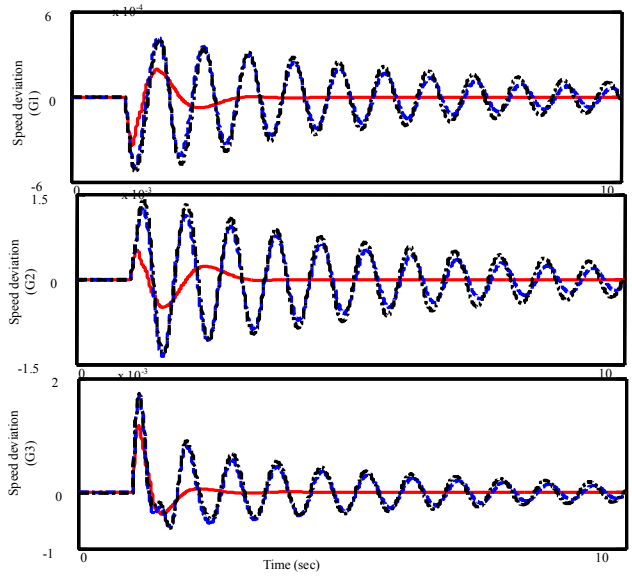


Figure 7. System response under heavy loading in scenario III; Solid (PSO-TVAC), Dashed (PSO), Dotted (CPSS)

It can be seen that the overshoot, undershoot, settling time and speed deviations of all machines are greatly reduced by applying the proposed PSO-TVAC PSSs.

IV. CONCLUSIONS

A robust PSO-TVAC PSS is proposed for the solution of the low frequency oscillation problem based on multi objective functions in multi machine power systems in this paper. The multi objective optimization problem has been solved by the PSO-TVAC optimization method. This method is stronger than other methods which considered with single objective in particular the lack of reliability in what concerns succeeded, and valid convergence, and the failures in attempts to reduce the time. The 3-machine 9-bus standard power system, under various system configurations and loading conditions, is employed to illustrate the performance of the proposed method. The effectiveness of the proposed controller is compared with the PSOPSS and CPSS through some performance indices. The simulation results indicate that this optimization method is very accurate and converges very rapidly so that it can be used in the practical optimization problems.

REFERENCES

[1] E.V. Larsen, D.A. Swann, "Applying power system stabilizers", Parts I–III, IEEE Trans. Power Apparatus Syst. Vol. 101, pp. 3017–3046, 1982.

[2] K. R. Padiyar, "Power system dynamics, stability and control". Second Ed., 2006.

[3] O. Abedinia, M. Salay Naderi, A. Jalili, B. Khamenehpour, "Optimal tuning of multi-machine power system stabilizer parameters using genetic-algorithm", Proceedings of International Conference on Power System Technology, Hangzhou, China, pp. 1-6, 24-28 Octobr, 2010.

[4] H. A. Shayanfar, H. Shayeghi, O. Abedinia, A. Jalili. "Design rule-base of fuzzy controller in multi-machine power system stabilizer using genetic algorithm". In: Proceedings of the international conference on artificial intelligence, July 12-14, Las Vegas, Nevada, USA, pp. 43-49, 2010.

[5] H.B. Duan, Wang D. B., ZHU J. Q., Huang X. H., "New development on ant colony algorithm theory and its applications". Control & Decision, Vol. 19, No. 12, pp. 1321 -1326, 2004.

[6] G. Ma, H, Duan, S. Liu, "Improved Ant Colony algorithm for global optimal trajectory planning of UAV under complex environment", International Journal of Computer Science & Applications, Vol. 4 Issue 3, pp 57-68, 2007.

[7] A. L. Barreiros, M. D. Ferreira, T. Costa-Jr, W. Barre-Jr, AP. Lopes, "A neural power system stabilizer trained using local linear controllers in a gain-scheduling scheme". Elect Power Energy Syst Vol. 27, pp. 473-479, 2005.

[8] M. Caner, N. Umurkan, S. Tokat, SV. Ustun, "Determination of optimal hierarchical fuzzy controller parameters according to loading condition with ANN", Expert Systems with Applications, Vol. 34, pp. 2650-2655, 2008.

[9] T. Niknam, "A new HBMO algorithm for multiobjective daily Volt/Var control in distribution systems considering distributed generators", Applied Energy, Vol. 88, pp. 778–788, 2011.

[10] E. Larsen, D. Swann," Applying power system stabilizers", IEEE Trans. Power App. Syst., Vol. 100, No. 6, PP. 3017- 3046, 1981.

[11] H. Shayeghi, H. A. Shayanfar, A. Safari, R. Aghmasheh, "A robust PSSs design using PSO in a multi-machine environment", Energy Conversion and Management, Vol. 51, pp. 1656-1670, 2010.

[12] AT. Al-Awami, YL. Abdel-Magid, MA. Abido "A particle-swarm based approach of power system stabilizer enhancement with unified power flow controller", Elect Power Energy Syst. Vol. 29, pp. 251-250, 2007.

[13] A. M. El-Zonkoly, A. A. Khalil and N. M. Ahmied, "Optimal tuning of lead-lag and fuzzy logic power system stabilizers using particle swarm optimizatio"", Expert System with Applications, Vol. 10, pp. 1-10, 2008.

[14] Y. Shi, R.C. Eberhart, "Empirical study of particle swarm optimization", In Proceeding of the Congress on Evolutionary Computation, Vol. 3, pp. 1945_1950, 1999.

[15] A. Ratnaweera, S. K. Halgamuge, H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients", IEEE Transactions on Evolutionary Computation, Vol. 8, No. 3, June 2004.

[16] A. Ratnaweera, S.K. Halgamuge, H.C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients", IEEE Transactions on Evolutionary Computation. Vol. 8, pp. 240_255, 2004.

BIOGRAPHIES

Ali Ghasemi received the B.S. degree in Electrical Engineering from Esfahan University of Technology, Esfahan, Iran in 2009. Currently, he is a M.S.E. student in Technical Eng. Department of the University of Mohaghegh Ardabili, Ardabil, Iran. His Areas of interest in Research are the Application of Heuristic Optimization to Power System Control.

Oveis Abedinia received the B.S. and M.Sc. degrees in Electrical Engineering from Azad University, Ardabil and Science and Technology Research Branch, Tehran, Iran in 2005 and 2009, respectively. His areas of interest in research are Application of Robust Control, Artificial Intelligence to Power System Control and restructuring, optimization in Distribution Generation and power system.

Heidarali Shayanfar received the B.S. and M.S.E. degrees in Electrical Engineering in 1973 and 1979, respectively. He received his Ph. D. degree in Electrical Engineering from Michigan State University, U.S.A., in 1981. Currently, he is a Full Professor in Electrical Engineering Department of Iran University of Science and Technology, Tehran, Iran. His research interests are in the Application of Artificial Intelligence to Power System Control Design, Dynamic Load Modeling, Power System Observability Studies, Voltage Collapse, Congestion Management in a Restructured Power System, Reliability Improvement in Distribution Systems and Reactive Pricing in Deregulated Power Systems. He has published more than 350 technical papers in the International Journals and Conferences proceedings. He is a member of Iranian Association of Electrical and Electronic Engineers and IEEE.

Mohammad Salay Naderi received his B.Sc. in electric power engineering (1996) and M.Sc. (1998) in high voltage engineering from Sharif University of Technology in Tehran, Iran. In 2006, he finished his PhD career at the same university which was on partial discharge monitoring and localization within power transformer. He joined Energy Systems Research Group in the School of Electrical Engineering and Telecommunications at UNSW in 2004 as Visiting Research Associate and continued his research until late 2005. He worked with Iran Power Generation and Transmission Company (Tavanir Holding Co.) as Deputy General Manager of Transmission System Technical Administration Bureau from 2005 to 2010. In 2010 he joined the School of Electrical Engineering and Telecommunications at UNSW as Lecturer. His research and teaching interests are in high voltage engineering and condition monitoring of power equipment; sustainable energy systems analysis and development as well as distributed generation and its effects of power system protection.

Design of Robust PSS to Improve Stability of Composed LFC and AVR Using ABC in Deregulated Environment

O. Abedinia
Islamic Azad University, Young Researcher Club, Ardabil Branch, Ardabil, Iran
E-mail: oveis.abedinia@hotmail.com

H. A. Shayanfar *
Electrical Eng. Department, Islamic Azad University, South Tehran Branch, Tehran, Iran
E-mail: hashayanfar@yahoo.com

B. Wyns
Department of Electrical Energy, Systems and Automation, Ghent, Belgium
E-mail: bart.wyns@ugent.be

A. Ghasemi
Technical Eng. Department, University of Mohaghegh Ardabili, Ardabil, Iran
E-mail: ghasemi.agm@gmail.com

Abstract— **This paper presents the robust design of a Power System Stabilizer (PSS) to improve the stability in the composed two-area Load Frequency Control (LFC) and Automatic Voltage Regulator (AVR) using Artificial Bee Colony (ABC) in a restructured environment. The coupling effects of the LFC and AVR loops are studied by extending the linearized Automatic Generation Control (AGC) system to include the excitation system. The proposed method is tested on the two-area power system. The simulation results show that adding a coordinative PSS on this model can improve the dynamic stability of the power system and effectively suppresses the low frequency oscillation. The effectiveness of the proposed method is compared with two-area system in a deregulated environment without AVR using Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) techniques.**

Keywords; Artificial Bee Colony, PSS, Load Frequency Control, AVR, Two-Area Power System.

I. INTRODUCTION

The objective of the control strategy in a power system is to generate and deliver power in an interconnected system as economically and reliably as possible while maintaining the frequency and voltage within permissible limits. The power system control has a hierarchical structure. The low-frequency oscillation of interconnected systems is one of the most important stability problems arising from large-scale electric power system interconnections [1].

Actually the complete model of a system in low-frequency oscillation study should consist of mechanical and electrical loops. There is no doubt that these oscillations can be controlled by adjusting exciter and speed-governor control parameters. Moreover, it has been shown that the load-voltage characteristic of the power system has a significant effect on its dynamic responses, and suggestions have been made for the proper representation of this characteristic in simulation studies [2]. On the other hand, Load Frequency Control (LFC) and Automatic Voltage Regulator (AVR) are two main control loops of a generation.

There are lots of papers studying LFC [3-5], however these researches didn't have any attention to the mutual effects between AVR and LFC. Also, these researches are based on the assumption that there is no interaction between power/frequency and the reactive-power/voltage control loops. However, there are some interactions between these two control channels in practical system during dynamic oscillations.

In [2] a PSS is added to combine with the LFC for more dynamic improvement but, the method of choosing appropriate PSS parameters have some problems. Over the years, several techniques have been developed for designing PSSs. Conventionally a lead-lag controller has been widely used in power system control to damp the low frequency oscillations. In order to determine the coefficient of the lead-lag controller, several control algorithms based on the conventional methods are proposed [6].

$H\infty$ optimization techniques [7-8] have been applied to the robust PSS design problem. However, the additive and/or multiplicative uncertainty representation cannot treat situations where a nominal stable system becomes unstable after being perturbed. On the other hand, the order of the $H\infty$ based stabilizer is as high as that of the plant. This gives rise to the complex structure of such stabilizers and reduces their applicability.

Particle Swarm Optimization (PSO) is one of the modern heuristic algorithms. It was developed through the simulation of a simplified social system, and has been found to be robust in solving continuous nonlinear optimization problems [9]. The PSO technique can generate a high quality solution within shorter calculation time and has a stable convergence characteristic than

* Correspanding Author. E-Mail Address: hashayanfar@yahoo.com (H. A. Shayanfar)

other stochastic methods [10]. Generally, PSO is characterized as a simple concept, easy to implement, and computationally efficient. Unlike the other heuristic techniques, PSO has a flexible and well-balanced mechanism to enhance the global and local exploration abilities. However, it is possible to observe that the PSO converges in local global. It is obvious that the conventional controllers which are optimized by PSO or GA have a suitable reaction in wide range systems. However, these optimizations of PSSs are valid [9] in a particular work point, whereas, this is not appropriate for various operating points.

In this paper, to overcome these problems, an Artificial Bee Colony (ABC) is proposed for the solution of tuning the PSS parameters. The ABC algorithm is a typical swarm-based approach to optimization, in which the search algorithm is inspired by the intelligent foraging behavior of a honey bee swarm process [11] and has emerged as a useful tool for the engineering optimization. It incorporates a flexible and well-balanced mechanism to adapt to the global and local exploration and exploitation abilities within a short computation time. Hence, this method is efficient in handling large and complex search spaces [12].

The effectiveness of the proposed method is tested on a two-area deregulated power system. The result of the proposed controller is compared with a GA Fuzzy (GAF) [13] and PSO Fuzzy (PSOF) [14] approaches in a restructured system through nonlinear time simulation and some performance indices. Evaluation results show that by using this model higher accuracy will be reached in the dynamic and steady state responses.

II. MODEL OF POWER SYSTEM

A. Load frequency Control (LFC)

Actually, the aim of the LFC is to maintain real power balance in the system through control of system frequency. Also, as the real power demand changes, a frequency change occurs. This frequency error is amplified, mixed and changed to a command signal which is sent to the turbine governor. The governor operates to restore the balance between the output and input by changing the turbine output. This strategy is also referred to as megawatt frequency or power-frequency (P-f) control [15].

B. Automatic Voltage Regulator (AVR)

The aim of this control is to maintain the system voltage between limits by adjusting the excitation of the machines. The AVR senses the difference between a rectified voltage derived from the stator voltage and a reference voltage. This error signal is amplified and fed to the excitation circuit. The change of excitation maintains the VAR balance in the network.

C. Structure of Combined LFC and AVR System

Deregulated power system consists of GENCOs, TRANSCOs and DISCOs with an open access policy. This is obvious that all transactions have to be cleared via Independent System Operator (ISO) or other responsible infrastructure. In this latter environment, it is appropriate that a new model for LFC scheme is improved to account for the effects of possible load following contracts on the system's dynamics [13].

According to the proposed idea in [4], the significant of an 'Augmented Generation Participation Matrix' (AGPM) to express the possible contracts following is presented here. The AGPM shows the communion factor of a GENCO in the load following contract with a DISCO. The dimension of the AGPM matrix in terms of

$$AGPM_{ij} = \begin{bmatrix} gpf_{(s_i+1)(z_j+1)} & \cdots & gpf_{(s_i+1)(z_j+m_j)} \\ \vdots & \ddots & \vdots \\ gpf_{(s_i+n_i)(z_j+1)} & \cdots & gpf_{(s_i+n_i)(z_j+m_j)} \end{bmatrix}$$

$$s_i = \sum_{k=1}^{i-1} n_i, z_j = \sum_{k=1}^{j-1} m_j, i, j = 2,..., N \;\&\; s_1 = z_1 = 0$$

rows and column is equal the total number of GENCOs and DISCOs in the overall power system, respectively. Consider the number of GENCOs and DISCOs in area i be ni and mi in a large scale power system with N control areas. The structure of the AGPM is given by:

$$AGPM = \begin{bmatrix} AGPM_{11} & \cdots & AGPM_{1N} \\ \vdots & \ddots & \vdots \\ AGPM_{N1} & \cdots & AGPM_{NN} \end{bmatrix} \quad (1)$$

Where, n_i and m_i define the number of GENCOs and DISCOs in area i and gpf_{ij} refers to the 'generation participation factor' and displays the participation factor of GENCO i in total load following requirement of DISCO j based on the possible contracts. The sum of all inputs in each column of AGPM is univalent. The block diagram of a generalized LFC model with AVR loop in a deregulated power system to control area i is presented in Fig.1. These new information signals are considered as disturbance channels for the decentralized LFC design. As there are many GENCOs in each area, ACE signal has to be distributed among them due to their ACE participation factor in the LFC task and $\sum_{j=1}^{mi} a\,pf_{ij} = 1$. It can be written that [5]:

$$d_i = \Delta P_{Locj} + \Delta P_{di} \quad , \quad \Delta P_{Locj} = \sum_{j=1}^{m_i} (\Delta P_{Lj-i} + \Delta P_{ULj-i}) \quad (2)$$

$$\eta_i = \sum_{j=1\,\&\,j\neq i}^{N} T_{ij}\Delta f_j, \quad \zeta_i = \sum_{k=1\,\&\,k\neq i}^{N} \Delta P_{tie,ik,sch}$$

$$\Delta P_{tie,ik,sch} = \sum_{j=1}^{n_i}\sum_{t=1}^{m_k} apf_{(s_i+j)(z_k+t)} \Delta P_{L(z_k+t)-k} - \sum_{t=1}^{n_k}\sum_{j=1}^{m_i} apf_{(s_k+t)(z_i+j)} \Delta P_{L(z_i+j)-i}$$

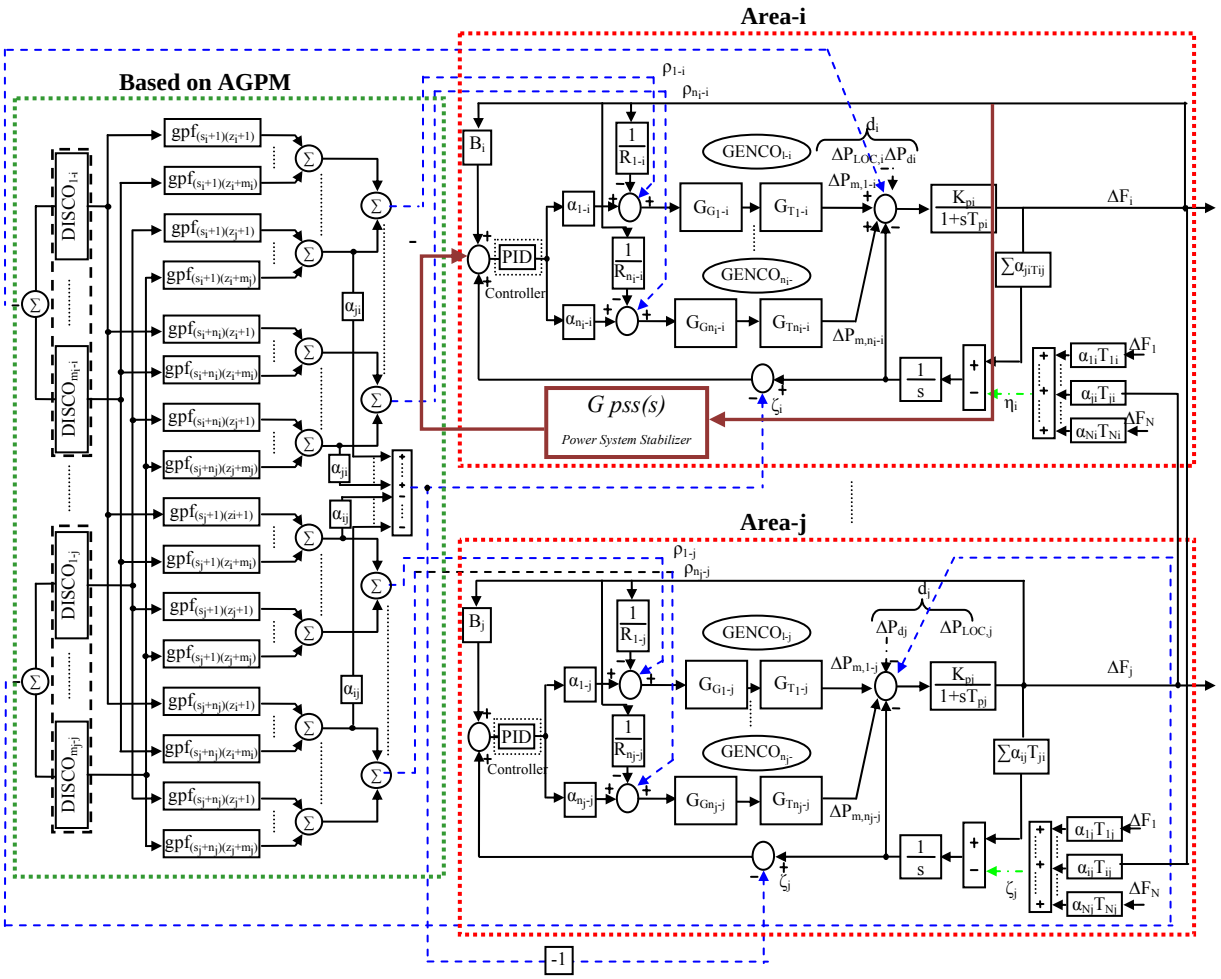


Figure 1. Generalized LFC model in the restructured system

$$\Delta P_{tie,i-error} = \Delta P_{tie,i-actual} - \zeta_i$$

$$\rho_i = [\rho_{1i} \ \cdots \ \rho_{ki} \ \cdots \ \rho_{ni}] \quad , \quad \rho_{ki} = \Delta P_{mki}$$

$$\Delta P_{m,k-i} = \sum_{j=1}^{z_{N+1}} gpf_{(s_i+k)j} \Delta P_{Lj-i} + apf_{ki} \sum_{j=1}^{m_i} \Delta P_{ULj-i}, k = 1,2,...,n_i$$

Where, $\Delta P_{m,ki}$ is the desired total power generation of a GENCO k in area i and must track the demand of the DISCOs in contract with it in the steady state. Two GENCOs and DISCOs are assumed to each control an area for which the system parameters are given in [3].

To make the visualization of contracts easier, the concept of a "DISCO Participation Matrix" (DPM) will be used. Essentially, DPM gives the participation of a DISCO in contract with a GENCO. In DPM, the number of rows has to be equal to the number of GENCOs and the number of columns has to the number of DISCOs in the system. Any entry of this matrix is a function of the total load power contracted by a DISCO toward a GENCO.

D. *Applied PSS for Proposed System*

The problem of setting the parameters of the PSSs that assure maximum damping performance is solved using an ABC algorithm. A widely used conventional lead-lag PSS is considered in this study which is shown in Fig. 2. Also a gain of K_s and four time constants T_1, T_2, T_3 and T_4 are considered for this controller.

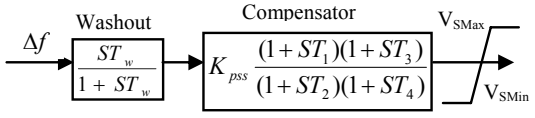


Figure 2. Power system stabilizer

E. *Artificial Bee Colony Algorithm*

The ABC algorithm is proposed by Karaboga [11] in 2005, and the performance of the ABC is analyzed in 2007 [16]. The foraging bees are classified into three categories; employed bees, onlookers and scout bees. All bees that are currently exploiting a food source are known as employed. The employed bees exploit the food source and they carry the information about the food source back to the hive and share this information with onlooker bees. Onlookers bees are waiting in the hive for the information to be shared by the employed bees about their discovered food sources and scout bees will always be searching for the new food sources near the hive. Employed bees share information about food sources by dancing in the

designated dance area inside the hive. The nature of this dance is proportional to the nectar content of the food source just exploited by the dancing bee. Onlooker bees watch the dance and choose a food source according to the probability proportional to the quality of that food source. Therefore, good food sources attract more onlooker bees compared to the bad ones. Whenever a food source is exploited fully, all the employed bees associated with it abandon the food source, and become scout. Scout bees can be visualized as performing the job of exploration, whereas, employed and onlooker bees can be visualized as performing the job of exploitation [12].

In the ABC algorithm, the number of employed bees is equal to the number of food sources which is also equal to the number of onlooker bees. There is only one employed bee for each food source whose first position is randomly generated. Each employed bee, at each iteration of the algorithm determines a new neighboring food source of its currently associated food source by the following equation, and computes the nectar amount of this new food source:

$$v_{ij} = z_{ij} + \theta_{ij}(z_{ij} - z_{kj})$$

Where, Θ_{ij} is a random number between [-1, 1]. If the nectar amount of this new food source is higher than that of its currently associated food source, then this employed bee moves to this new food source, otherwise it continues with the old one. After all employed bees complete the search process; they share the information about their food sources with onlooker bees. An onlooker bee evaluates the nectar information taken from all employed bees and chooses a food source with a probability related to its nectar amount by this equation. This method, known as roulette wheel selection method, provides better candidates to have a greater chance of being selected:

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_i}$$

Where, fit_i is the fitness value of the solution i which is proportional to the nectar amount of the food source in the position i and SN is the number of food sources which is equal to the number of employed bees. Pseudo code for ABC algorithm:

- Initialize
- Repeat.
- Move the employed bees onto their food source and evaluate the fitness
- Move the onlookers onto the food source and evaluate their fitness
- Move the scouts for searching new food source
- Memorize the best food source found so far
- Until (termination criteria satisfied)

F. *Proposed ABC-PSS in LFC*

The proposed controller was applied for one area in a two-area LFC power system in the deregulated environment. Also, the proposed algorithm optimizes the parameters of the LFC power system as K_{Pi}, K_{Ii} and K_{Di} for i=1, 2… N. For both of the PSS and LFC optimization these "Time multiplied Absolute value of the Error" ITAE (Δf) and ITAE (ΔP) are calculated for the objective functions which are defined as:

$$ITAE(\Delta f) = 150 \times \int_0^{tsim} |dF1| dt$$

$$ITAE(\Delta P) = 150 \times \int_0^{tsim} |dp12| dt$$

Where, the constraints of the PID controller parameter bounds which are considered with PSS parameters consist of:

$$\text{Minimize } u \text{ subject to}: \quad \begin{aligned} K_{Pi}^{\min} &\le K_{Pi} \le K_{Pi}^{\max} \\ K_{Ii}^{\min} &\le K_{Ii} \le K_{Ii}^{\max} \\ K_{Di}^{\min} &\le K_{Di} \le K_{Di}^{\max} \end{aligned}$$

Also the bounds of PSS parameters are presented in Table 1.

TABLE I. PSS LIMITATIONS

Parameters	T_1	T_2	T_3	T_4	K_{pss}
Lower limit	0.01	0.01	0.01	0.01	0.01
Uper limit	3	3	3	3	3

Moreover, the trend of the objective function of the algorithm is presented in Fig. 3. Also, the optimized parameters of the PID and PSS are presented in Table. 2 and 3, respectively.

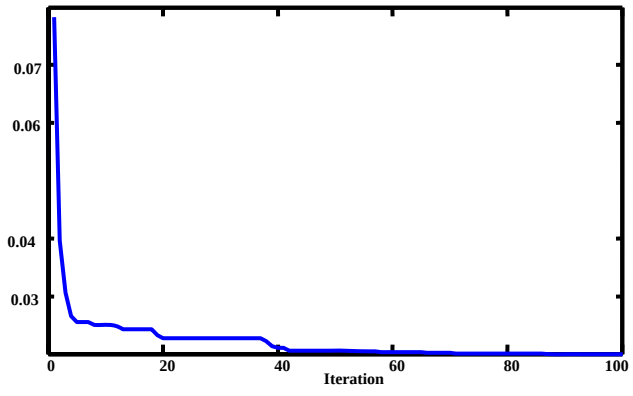


Figure 3. Variations of fitness function.

TABLE II. OPTIMUM PID CONTROLLER GAINS FOUND BY ABC

Algorithm	K_{p1}	K_{i1}	K_{d1}	K_{p2}	K_{i2}	K_{d2}
ABC	2.7634	2.9929	1.3690	2.8406	0.3751	1.5638

TABLE III. OPTIMUM PSS CONTROLLER PARAMETERS FOUND BY ABC

Algorithm	K_{pss}	T_1	T_2	T_3	T_4
ABC	0.0126	0.2293	3.4321	1.6034	1.6539

III. SIMULATION Results

In this part, a PSS is designed to display the dynamic improvement for a composed model of LFC and AVR. The simulation is performed for different possible operating conditions of the power system.

It is supposed that each DISCO demands 0.1 pu MW power from GENCOs. Also, it is possible that a DISCO violates a contract by demanding more power than the amount specified in the contract. This additional power must be reflected as a local load of the area but not as the contract demand and taken up by the GENCOs in the same area. Moreover, DISCOs 1, 2 each demand 0.05 puMW of additional power. According to equation (2) the total loads in areas are calculated as:

$$P_{Loc,1}=\Delta P_{L,1\text{-}1}+\Delta P_{L,2\text{-}1}+\Delta P_{U,1\text{-}2}=0.1+0.1+0.05=0.25$$

$$P_{Loc,2}=\Delta P_{L,1\text{-}2}\Delta P_{L,2\text{-}2}\Delta P_{U,1\text{-}2}=0.1+0.1+0.05=0.25$$

To make the visualization of contracts easier, the concept of a "DISCO Participation Matrix" (DPM) will be used. Essentially, DPM gives the participation of a DISCO in contract with a GENCO. The purpose of this scenario is to test the effectiveness of the proposed controller against uncertainties and large load disturbances in the presence of Generation Rate Constraints (GRC). For this purpose the elements of the DPM are cpf_{ij} the factor for restructured system as:

$$DPM = \begin{bmatrix} 0.5 & 0.25 & 0 & 0.3 \\ 0.2 & 0.25 & 0 & 0 \\ 0 & 0.25 & 1 & 0.7 \\ 0.3 & 0.25 & 0 & 0 \end{bmatrix}$$

The results of the proposed controller are compared with the PSO-F PID [14] and GA-F PID [13]. Also, disturbances of the area are the following: dP1=0.05, dP2=0.03. Considering to this scenario the modifications of excess power as: Disco1=0.17 (equal to 0.1+0.07), Disco2=0.14 (equal to 0.1+0.04), Disco3=0.12 (equal to 0.1+0.02), Disco4=0.12 (equal to 0.1+0.02). Also, the factors of PID controller are: apf1=0.75, apf2=1-apf1, apf3=.5, apf4=1-apf3. The results for +25% and -25% changes of parameters for the LFC are shown in Fig. 4-5, respectively.

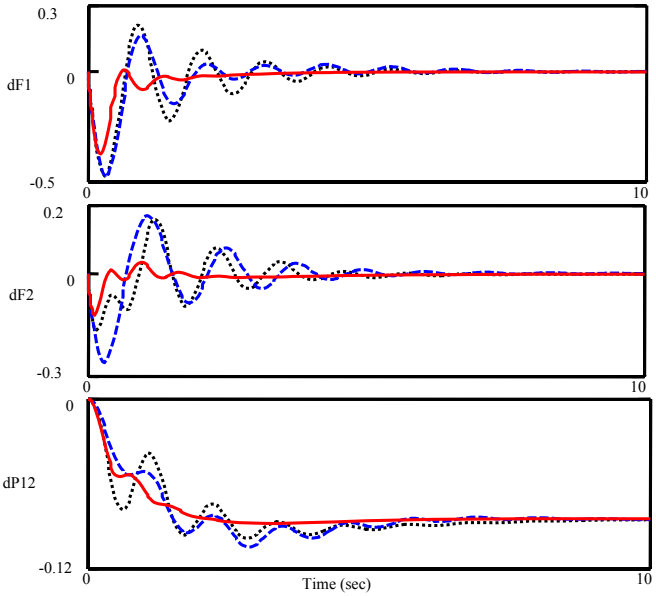


Figure 4. Deviation of frequency and tie lines power flows for +25% changes; Solid (ABC with PSS), Dashed (PSO-F without PSS) and Dotted (GA-F without PSS).

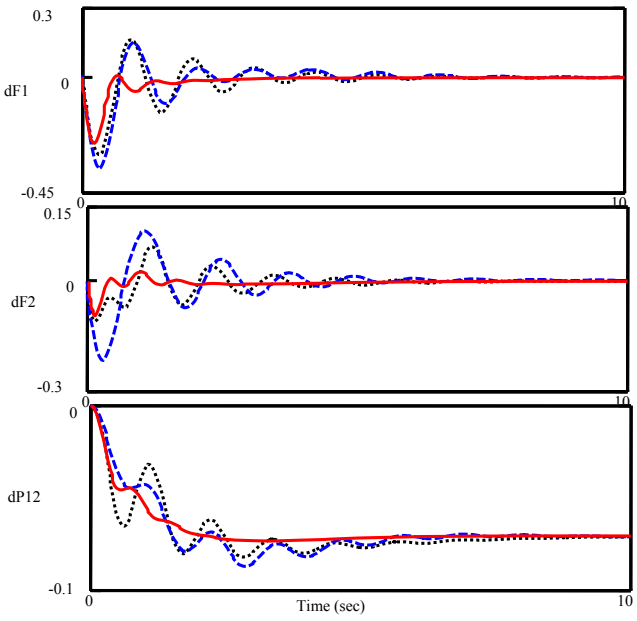


Figure 5. Deviation of frequency and tie lines power flows for -25% changes; Solid (ABC with PSS), Dashed (PSO-F without PSS) and Dotted (GA-F without PSS).

The simulation results represent the positive effect of the AVR composed LFC on the improvement of the oscillation of frequency due to any load demands and disturbances. It can be seen that the proposed control strategy can ensure robust performance such as possible contracted scenario under modeling uncertainties in the various operating conditions.

IV. CONCLUSION

In this research, a new combination of LFC and AVR based ABC technique was introduced for improvement of stability in power system in a restructured environment. In this paper the effect of the AVR loop is considered over LFC loop which leads to an improvement in the mentioned power system. The parameters of PSS and PID controller are tuned according to some performance index based ABC. The ABC algorithm is a search algorithm that is inspired by the intelligent foraging behavior of a honey bee swarm process and has emerged as a useful tool for engineering optimization. It incorporates a flexible and well-balanced mechanism to adapt to the global and local exploration and exploitation abilities within a short computation time. It is well known that the conventional methods to tune the gains of the PID controller and PSS parameters with numerical analysis may be tedious and time consuming. This control strategy was chosen because of the increasing complexity and changing structure of the power systems. The effectiveness of the proposed method is tested on a two-area restructured power system for a wide range of load demands and disturbances under different operating conditions that is compared via PSOF, GAF controllers through some performance indicates and could be ensure the robust performance such for possible contracted scenario under modeling uncertainties in the various operating conditions.

REFERENCES

[1] E. Rakhshani, J. Sadeh, "Application of power system stabilizer in a combined model of LFC and AVR loops to enhance system stability", International Conference on Power System Technology (IEEE), No. 5940, pp. 1-5, 2010.

[2] E. Rakhshani, K. Rouzbehi, S. Sadeh, "A new combined model for simulation of mutual effects between LFC and AVR loops", In Proc. Asia-Pacific Power and Energy Engineering Con., Vol. 2, pp. 1480-1485, March. 2009.

[3] F. Liu, YH. Song, J. Ma, Q. Lu. "Optimal load frequency control in the restructured power systems". IEE Proc Gener Transm Distrib, Vol. 15, No. 1, 2003.

[4] H. Shayeghi, H. A. Shayanfar, OP. Malik, "Robust decentralized neural networks based LFC in a deregulated power system", Elect Power Syst Res, Vol. 77, No. 3, pp. 241–51, 2007.

[5] H. Shayeghi, H. A. Shayanfar. "Decentralized load frequency control of a restructured electric power system using ANN technique". WSEAS Trans Cir Syst, pp. 38–47, 2005.

[6] O. Abedinia, M. Salay Naderi, A. Jalili, B. Khamenehpour, "Optimal tuning of multi-machine power system stabilizer parameters using genetic-algorithm", Proceedings of International Conference on Power System Technology, pp. 1-6, Hangzhou, China, 24-28 Octobr, 2010.

[7] J. H. Hwang, D. W. Kim, J. H. Lee, Y. J. An, "Design of fuzzy power system using adaptive evolutionary algorithm", Engineering Applications of Artificial Intelligence, Vol. 21, pp. 86-96, 2008.

[8] H. A. Shayanfar, H. Shayeghi, O. Abedinia, A. Jalili. "Design rule-base of fuzzy controller in multi-machine power system stabilizer using genetic algorithm". In: Proceedings of the International Conference on Artificial Intelligence, July 12-14, Las Vegas, Nevada, USA, pp. 43-49, 2010.

[9] AT. Al-Awami, YL. Abdel-Magid, MA. Abido. "A particle-swarm based approach of power system stabilizer enhancement with unified power flow controller", Elect Power Energy Syst. Vol. 29, pp. 251-250, 2007.

[10] H. Shayeghi, H. A. Shayanfar, A. Safari, R. Aghmasheh, "A robust PSSs design using pso in a multi-machine environment", Energy Conversion and Management, Vol. 51, pp. 1656-1670, 2010.

[11] D. Karaboga, "An idea based on honey bee swarm for numerical optimization", Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[12] SL. Sabat, SK. Udgata, A. Abraham, "Artificial bee colony algorithm for small signal model parameter extraction of MESFET", Engineering Applications of Artificial Intelligence, pp. 1-6, 2010.

[13] H. A. Shayanfar, H. Shayeghi, A. Jalili, M. Sivandian, "A genetic algorithm based AGC of a restructured power system", In: Proceedings of the International Conference on Artificial Intelligence, Las Vegas, Nevada, USA, pp. 237-240, June 2006.

[14] H. Shayeghi, H. A. Shayanfar, "PSO based neuro-fuzzy controller for LFC design including communication time delays", International Journal on TPE (Technical and Physical Problems of Engineering) IJTPE, Vol. 1, No. 2, pp. 28-36, June 2010.

[15] A. Soundarrajan, Dr. S. Sumathi, C. Sundar, "Particle swarm optimization based LFC and AVR of autonomous power generating system", IAENG International Journal of Computer Science, Vol. 37, No. 1, 2010

[16] D. Karaboga, B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial Bee Colony (ABC) algorithm", Journal of Global Optimization, Springer Netherlands, Vol. 39, No.3, pp.459-471, 2007.

BIOGRAPHIES

Oveis Abedinia received the B.S. and M.Sc. degrees in Electrical Engineering from Azad University, Ardabil and Science and Technology Research Branch, Tehran, Iran in 2005 and 2009, respectively. His areas of interest in research are the Application of Robust Control, Artificial Intelligence to Power System Control and restructuring, optimization in Distribution Generation and power system.

Heidarali Shayanfar received the B.S. and M.S.E. degrees in Electrical Engineering in 1973 and 1979, respectively. He received his Ph. D. degree in Electrical Engineering from Michigan State University, U.S.A., in 1981. Currently, he is a Full Professor in Electrical Engineering Department of Iran University of Science and Technology, Tehran, Iran. His research interests are in the Application of Artificial Intelligence to Power System Control Design, Dynamic Load Modeling, Power System Observability Studies, Voltage Collapse, Congestion Management in a Restructured Power System, Reliability Improvement in

Distribution Systems and Reactive Pricing in Deregulated Power Systems. He has published more than 350 technical papers in the International Journals and Conferences proceedings. He is a member of Iranian Association of Electrical and Electronic Engineers and IEEE.

Bart Wyns received the B.S. and M.Sc. degree in Computer Science in 1999 and 2001, respectively from Ghent University, Belgium. He received his Ph.D. degree in Computer Science from the same university in 2007. Currently, he is a postdoctoral research fellow in the Department of Electrical Energy, Systems and Automation. His research interests include application of artificial intelligence to mobile robotics, swarm robotics, human machine interfacing.

Ali Ghasemi received the B.S. degree in Electrical Engineering from Esfahan University of Technology, Esfahan, Iran in 2009. Currently, he is a M.S.E. student in Technical Eng. Department of the University of Mohaghegh Ardabili, Ardabil, Iran. His Areas of interest in Research are the Application of Heuristic Optimization to Power System Control.

An Automatic Image Registration Algorithm for Tracking Moving Objects in Low-Resolution Video

David O. Johnson[1] **and Arvin Agah**[2]
[1]Computer Science and Electrical Engineering, University of Missouri – Kansas City, Kansas City, MO, USA
[2]Electrical Engineering and Computer Science, University of Kansas, Lawrence, KS, USA

Abstract - *We propose an automatic image registration algorithm for tracking moving objects in low-resolution videos. The algorithm uses SIFT keypoints to identify matching stationary points in the input frame and the base frame. The best set of matching stationary points is used to create a spatial transform to register the points on the moving objects in the input frame to the base frame. We examined two probabilistic methods and one deterministic method of identifying the stationary points with two different fitness measures (Euclidean and Mahalanobis) and two spatial transforms (affine and projective). Our experiments on two low-resolution videos indicate that our algorithm performs better using the affine transform than the projective transform. However, the differences in average pixel error between the methods of determining stationary points and fitness measures are statistically insignificant. Therefore, which one to use depends on the execution speed and confidence interval required for the application.*

Keywords: video analysis; robotic; motion analysis; tracking objects; Scale Invariant Feature Transform (SIFT); image registration

1 Introduction

Today there are no general-purpose service robots that can perform household activities. There are special purpose machines that can vacuum a room (e.g., the Roomba commercial robot [7]). However, only robots in laboratory environments can do so by manipulating a vacuum cleaner. It is envisioned that someday there will be general-purpose service robots that can perform these types of tasks. These general-purpose robots will need to be programmed to perform all kinds of human activities. The organizers of the Interactive Robot Learning Workshop, Robotics: Science and Systems 2008, held in Zurich, Switzerland in June, 2008 [14] stated: "Many future applications for autonomous robots bring them into human environments as helpful assistants to untrained users in homes, offices, hospitals, and more. These applications will often require robots to flexibly adapt to the dynamic needs of human users. Rather than being pre-programmed at the factory with a fixed repertoire of skills, these personal robots will need to be able to quickly learn how to perform new tasks and skills from natural human instruction. Moreover, it is our belief that people should not have to learn a new form of interaction in order to teach these machines, that the robots should be able to take advantage of communication channels that are natural and intuitive for the human partner." In other words, instead of programming these robots, we should teach them in the same way humans teach each other. One cost-effective method of teaching is using training videos, as there are numerous training videos for teaching various types of skills. One challenge associated with using training videos is how does the vision system detect physical movements in an "untethered", or markerless, environment, where the user does not wear special clothing or sensors? Inamura *et al*. used Hidden Markov Models to encode human joint trajectories into proto-symbols [5, 6]. In their experiments, they were able to show that a HRP-2W service robot could learn and demonstrate the proto-symbol by watching a human perform it. However, the joints of the human teacher were tracked using a wearable motion capturing system, which will not be likely in training videos. In this paper, we propose an automatic image registration algorithm for tracking moving objects in low-resolution videos (240 pixels wide and 180 pixels high), which when combined with the object detection scheme we used essentially solves the "untethered" problem.

We found that using the affine transform with our algorithm produces better results (an average error of 1 to 2 pixels) than the projective transform. However, the differences in average pixel error between the methods of determining stationary points and fitness measures is statistically insignificant, particularly in light of the fact that the maximum video resolution is only one pixel. Therefore, which ones to use with our algorithm depends on the execution speed and confidence interval required for the application.

This paper is organized into five sections. Section 2 discusses the related work. Section 3 presents the theory of our automatic image registration algorithm for tracking moving objects in low-resolution videos. Section 4

discusses the results of the experiments that we performed to verify the algorithm. Section 5 concludes the paper.

2 Related work

Various methods of video tracking and pose estimation have been proposed. Some of the more recent methods are discussed here. Deutscher and Reid used a modified particle filter, which they call annealed particle filtering to recover full articulated body motion from markerless human motion [2]. John *et al*. used a hierarchical search algorithm to match human models of truncated cones with edges and silhouettes detected from multiple cameras to track the three-dimensional pose of a human [8]. Both of these methods rely on multiple cameras which would not work with single camera training videos. Ong *et al*. used a particle filter to match exemplars of the human skeleton with edges of limbs detected from an image to track the three-dimensional pose of a human across video frames from a single camera [11]. Chen and Schonfeld propose a method to track the object's motion and estimate its pose directly from two-dimensional image sequences using Lowe's Scale-Invariant Feature Transform (SIFT) [1, 9]. In our work, we want to use machine learning to recognize a sequence of n-dimensional vectors of x-y coordinates of inanimate object and body part centroids as a specific proto-symbol. Video tracking and pose estimation, on the other hand, seek to extract features from the image (e.g., edges, silhouettes, SIFT keypoints) and perform a best match to known models of the human body in particular poses. Thus, the output is a sequence of body poses that can be replicated for animation, virtual reality, etc., or as an alternative to our method of representing poses as a sequence of body part centroids.

Various methods of image registration have been proposed. Zitova and Flusser give a detailed overview of registration techniques from a general point of view [17]. Wyawahare *et al*. provide a more recent overview of image registration techniques used in medical imaging, which are also generally applicable [16]. Silveira and Malis provide an excellent summary of the latest advances in image registration and propose a new photo-geometric transformation model and optimization methods for directly registering color and black-and-white images [13]. Then, they show that widely adopted models are in fact specific cases of their proposed model. They showed their method works well at restoring still images that have been warped and relit back to an original reference image. The methods discussed by Zitova and Flusser, Wyawahare *et al*., and Silveria and Malis register all pixels of a still image, where we are only interested in registering a few points of a moving image.

Closer to our approach, Sheikh *et al*., used image registration in a background subtraction algorithm to identify the parts of a scene that are stationary [18]. The significant differences between their work and ours is (1) we tested two other algorithms for determining the stationary points, in addition to RANSAC, which they used; (2) we tested using the projective transform, in addition to the affine transform, which they used; (3) we tested the Mahalanobis distance as a fitness measure, in addition to the Euclidean distance, which they used; (4) we used SIFT keypoints to identify common points between two images, where they used a particle filter; (5) their images were higher resolution; and (6) we tested with14 objects in two different videos and they tested with 1 object in 3 videos.

3 Image registration algorithm

The problem with finding the trajectories of points on moving objects in a video is that the camera moves and zooms in and out. Thus, the coordinate system of each frame cannot be used to define the trajectory of a point. Image registration can be used to translate the trajectories of the points in each frame to a common coordinate system. Each frame represents an image of the scene taken from a different viewpoint. Image registration is the process of aligning two or more images of the same scene [10]. Typically, one image, called the base image or reference image, is considered the reference to which the other images, called input images, are compared. The object of image registration is to bring the input image into alignment with the base image by applying a spatial transformation to the input image. A spatial transformation maps locations in one image to new locations in another image. Determining the parameters of the spatial transformation needed to bring the images into alignment is key to the image registration process. The parameters of the spatial transform can be determined by finding a small set of matching stationary points in each image. The required number of matching stationary points depends on the spatial transform. For example, an affine transform requires a minimum of three matching points and a projective transform requires four matching points. The spatial transform is then applied to the non-stationary points in the input frame to translate them to the coordinate system of the base frame. The result is the trajectories of the non-stationary points in the coordinate system of the base frame.

Lowe's SIFT keypoints are an excellent method of identifying matching points between two images of a scene (or frames of a video) [9]. The image registration problem is then reduced to determining which of the matching points are stationary and which are non-stationary. Once

the matching stationary points are determined, the only question left to answer, is which spatial transform to use. We looked at three methods of identifying the stationary points with two different fitness measures and two spatial transforms, which are explained below.

3.1 Identifying stationary points

We looked at two probabilistic algorithms for determining the stationary points: RANSAC [3] and SURSAC [12], and one deterministic algorithm, which we call DSPPE (Determining Stationary Points by Process of Elimination).

RANSAC is an iterative method to estimate parameters of a mathematical model from a set of observed data which contains outliers and inliers. The inliers fit the mathematical model and the outliers do not. For our problem, the inliers are the stationary points, the outliers are the non-stationary points, and the model is the spatial transform, which will register the input frame to the base frame. The maximum number of iterations, *k*, is calculated by the formula:

$$k = \log(1 - p)/\log(1 - w^s) \quad (1)$$

where,

p = probability that algorithm produces a useful result
w = number of inliers in data / number of points in data
s = minimum number of data required to fit the model

RANSAC assumes the ratio of inliers to data points, *w*, is known a priori, which is not the case in our application. So, we used a version of RANSAC proposed by Hartley and Zisserman [4], which Scherer-Negenborn and Schaefer called RANADAPT [12]. RANADAPT is different from RANSAC in that *w* is estimated each iteration by dividing the maximum number of inliers found so far by the number of data points. RANADAPT runs until the number of iterations exceeds the calculated number of expected iterations, *k*, which decreases each time a bigger inlier set is found. The RANADAPT algorithm adapted to finding stationary points is shown in Fig. 1.

Scherer-Negenborn and Schaefer cited other work which observed that the number of iterations, *k*, required to derive good model parameter values used by RANSAC-like model estimators is too optimistic [12]. They proposed an improvement to RANSAC, called Sufficient Random SAmple Coverage (SURSAC), which corrects this deficiency. SURSAC calculates k as follows:

$$k = \log(1 - p)/\log(1 - p_m) \quad (2)$$

where,

p = probability that algorithm produces a useful result
$p_m = w(s!/s^s)$
w = number of inliers in data / number of points in data
s = minimum number of data required to fit the model

To implement SURSAC, we modified the algorithm in Fig. 1 to calculate *k* as described above.

DSPPE is a deterministic algorithm that starts by assuming all the data points are inliers. The error in translating the input frame points to the base frame is calculated using the spatial transform fitted to all of the inliers. Then, the inlier point that reduces the error the most is removed from the inlier set. The process of removing the point that reduces the error the most is repeated until the inlier set with the minimum error is found. The DSPPE algorithm is shown in Fig. 2.

```
input:
   data - set of matched SIFT points from base and input frame
   s - minimum number of points for the spatial transform
   model - spatial transform that can be fitted to data
   t - threshold value for determining when a datum fits a model
   d - number of close data values to assert that model fits data
   p - probability that algorithm produces useful result
output:
   best_model – spatial transform which best fits the data

iterations := 0
best_model := nil
best_error := infinity
largest_inlier_set_sz := s
n := number of data points
w := largest_inlier_set_sz/n
k := log(1-p)/log(1-w^s)
while iterations < k
   maybe_inliers := s randomly selected values from data
   maybe_model := model parameters fitted to maybe_inliers
   consensus_set := maybe_inliers
   for every point in data not in maybe_inliers
      if point fits maybe_model with an error smaller than t
         add point to consensus_set
   if the number of elements in consensus_set is > d
      better_model := model parameters fitted to all points in
consensus_set
      this_error := a measure of how well better_model fits data
      if this_error < best_error
         best_model := better_model
         best_error := this_error
         if the size of the consensus_set > largest_inlier_set_sz
            largest_inlier_set_sz := size of consensus_set
            w := largest_inlier_set_sz/n
            k := log(1-p)/log(1-w^s)
   increment iterations
 return best_model
```

Figure 1. RANADAPT algorithm for determining the best spatial transform.

```
input:
   data - set of matched SIFT points from base and input  frame
   s - minimum number of points for the spatial transform
   model - spatial transform that can be fitted to data
output:
   best_model – spatial transform which best fits the data

maybe_inliers := all data points
maybe_model := model parameters fitted to maybe_inliers
best_error := a measure of how well maybe_model fits data
best_inliers_this_round := maybe_inliers
best_model := maybe_model
while the number of maybe_inliers > s
   best_error_this_round := infinity
   best_inliers_last_round := best_inliers_this_round
   for each point in best_inliers_last_round
      maybe_inliers := best_inliers_last_round, 1 point removed
      maybe_model := model parameters fitted to maybe_inliers
      this_error := a measure of how well maybe_model fits data
      if this_error < best_error_this_round
         best_error_this_round := this_error
         best_inliers_this_round := maybe_inliers
         if this_error < best_error
            best_error := this_error
            best_model := maybe_model
return best_model
```

Figure 2. DSPPE algorithm for determining the best spatial transform.

3.2 Fitness measures

The fitness measure (e.g., *this_error*) is a measure of how well a proposed model (i.e., spatial transform) fits the data. It is calculated by first using the proposed spatial transform to translate the input frame matching points to the base frame, and then comparing the translated points with their matching points in the base frame. Theoretically, if the spatial transform is created from only stationary points, then the only error will be due to the non-stationary points, and therefore should be less than any error arising from a spatial transform created from a mix of stationary and non-stationary points.

Mathematically, this is represented as follows:

$$E(\mathbf{I'},\mathbf{B}) = \text{error between } \mathbf{I'} \text{ and } \mathbf{B} \qquad (3)$$

where,

B = matching points in base frame
I = matching points in input frame
I' = S(**I**) = matching points in input frame translated to base frame coordinate system
S = spatial transform

We looked at two fitness measures: the Euclidean distance between **I'** and **B**, and the Mahalanobis distance between them.

3.3 Spatial transforms

We looked at two spatial transforms: affine and projective. The affine transform is used when shapes in the input image exhibit shearing [10]. Straight lines remain straight, and parallel lines remain parallel, but rectangles become parallelograms. The affine transform requires a minimum of three pairs of matching points. The projective transform is used when the scene appears tilted. Straight lines remain straight, but parallel lines converge toward vanishing points that might or might not fall within the image. The projective transform requires a minimum of four pairs of matching points. When more than the minimum the number of matching pairs was available, we used a least squares solution.

4 Results

To measure the effectiveness of our algorithm, we created two sets of unregistered trajectories from two different low-resolution (240 pixels wide and 180 pixels high) instructional videos on cleaning golf clubs. We used Lowe's SIFT keypoints with a nearest neighbor object detection approach to track the centroids of 14 objects across 40 frames of each video [19]. Fig. 3 illustrates the centroids of the 14 objects detected in Video 1.

Figure 3. Centroids of the 14 objects detected in one frame of Video 1.

Although we used the trajectories of centroids, our method could be used to register trajectories of points in other object representations, such as the vertices in a wire-frame representation. We then manually selected a base frame in each video to register the other 39 frames to. Although we selected the base frame manually, it could have been done automatically by selecting the frame that had the most SIFT keypoints in common with the other 39 frames. We then applied our algorithm to the 39 other input frames to obtain a spatial transform to register that frame to the base frame. We then applied the spatial transform to the centroids detected in that frame to create the registered

trajectory. We then measured the average pixel error between the registered trajectory and a ground-truth trajectory, as follows:

$$e = (\sum(\boldsymbol{R}_{ij} - \boldsymbol{G}_{ij})2)½/C_k \quad (4)$$

where,

- e = average pixel error
- $\boldsymbol{R}_{ij}$ = vector of x and y coordinates of registered centroids for object i in frame j
- $\boldsymbol{G}_{ij}$ = vector of x and y coordinates of registered ground-truth centroids for object i in frame j
- C_k = the number of registered centroids in video k, excluding the base frame
- C_1 = 134 for video 1
- C_2 = 176 for video 2

The registered ground-truth centroids were obtained in two steps. First, the unregistered ground-truth centroids were calculated by manually drawing borders around the objects in each frame and then taking the average of the coordinates of the pixels contained within the drawn borders. Second, the unregistered ground-truth centroids were registered to the base frame by manually selecting four matching stationary points to create a projective transform. Fig. 4 shows the trajectory of the registered ground-truth centroids for the towel in Video 2.

Figure 4. Blue dots and lines show trajectory of the registered ground-truth centroids for the towel in Video 2.

For the RANSAC and SURSAC algorithm parameters we used: t (threshold value for determining when a datum fits a model) = 10 pixels; d (number of close data values to assert that model fits data) = s; and p (probability that algorithm produces a useful result) = 99.9%. The experiments were implemented using the MATLAB Image Processing Toolbox [10] and the VLFeat open source library of computer vision algorithms [15].

Table 1 shows the average pixel error, 95% confidence interval, and relative execution speed for each of the combinations of methods of identifying the stationary points, fitness measures, and spatial transforms that we tested. The average pixel error results are also illustrated in Fig. 5.

5 Conclusion

The average pixel error and confidence intervals of Video 2 are better than Video 1 because the camera movement in Video 2 was less.

The results indicate that our algorithm performs better using the affine transform than the projective transform. However, the differences in average pixel error between the methods of determining stationary points (i.e., DSPPE, RANSAC, and SURSAC) and fitness measures (i.e., Euclidean and Mahalanobis) is statistically insignificant, particularly in light of the fact that the maximum video resolution is only one pixel. Therefore, which one to use depends on the execution speed and confidence interval required for the application. In other words, our algorithm performs the best (as measured by average pixel error) when the affine transform is used, but it does not matter which method of determining stationary points (i.e., DSPPE, RANSAC, or SURSAC) or fitness measure (i.e., Euclidean or Mahalanobis) is used.

TABLE I. EXPERIMENTAL RESULTS

	Average Pixel Error		95% Confidence Interval		Rel. Exe. Time
	Video 1	Video 2	Video 1	Video 2	
RANSAC (Aff, Euc)	1.83	1.07	0.03	0.03	1
SURSAC (Aff, Euc)	1.91	1.13	0.06	0.02	4
DSPPE (Aff, Euc)	1.96	1.02	0.00	0.00	2
RANSAC (Aff, Mah)	2.05	1.14	0.02	0.06	1
DSPPE (Aff, Mah)	2.09	1.11	0.00	0.00	2
SURSAC (Aff, Mah)	2.10	1.19	0.13	0.14	4
DSPPE (Pro, Euc)	3.30	2.20	0.00	0.00	2
SURSAC (Pro, Euc)	11.98	1.89	4.61	0.42	17
RANSAC (Pro, Euc)	12.28	1.79	3.03	0.07	2

The automatic image registration algorithm proposed in this paper combined with the object detection scheme we used [19] essentially solves the "untethered" problem associated with using instructional videos to program robots by demonstration. The automatic image registration

algorithm for tracking moving objects in low-resolution videos that we propose here can also be used in many other applications to "register" the trajectory of moving points in a video.

6 References

[1] C. Chen and D. Schonfeld (2010). A Particle Filtering Framework for Joint Video Tracking and Pose Estimation. IEEE Transactions on Image Processing, 19(6), 1625-1634.

[2] J. Deutscher and I. Reid (2005). Articulated Body Motion Capture by Stochastic Search. International Journal of Computer Vision, 61(2), 185-205.

[3] M. Fischler and R. Bolles (1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Communications of the ACM, 24(6), 381–395.

[4] R. Hartley and A. Zisserman (2003). Multiple View Geometry. In Computer Vision (2nd ed.). New York, Cambridge University Press.

[5] T. Inamura, K. Okadab, S. Tokutsub, N. Hataob, M. Inabab, and H. Inoue (2006). HRP-2W: A Humanoid Platform for Research on Support Behavior in Daily Life Environments. In Proceedings of the 9th International Conference on Intelligent Autonomous Systems (IAS-9), Tokyo, Japan, March 7- 9, 2006, 57(2), 145-154.

[6] T. Inamura, I. Toshima, and Y. Nakamura (2003). Acquiring Motion Elements for Bidirectional Computation of Motion Recognition and Generation. In B. Siciliano and P. Dario (Eds.) Experimental Robotics VIII, Springer-Verlag, Vol. 5, 372-381.

[7] iRobot Corporation Website (2010). www.irobot.com. Accessed 31 October 2010.

[8] V. John, E. Trucco, and S. Ivekovic (2010). Markerless Human Articulated Tracking Using Hierarchical Particle Swarm Optimization. Image and Vision Computing, 28(11), 1530-1547.

[9] D. Lowe (1999). Object Recognition from Local Scale-Invariant Features. In Proceedings of the International Conference on Computer Vision, 2, 1150–1157.

[10] The MATLAB Image Processing Toolbox 6.3 (2009). MathWorks. www.mathworks.com.

[11] E. Ong, A. Micilotta, R. Bowden, and A. Hilton (2006). Viewpoint Invariant Exemplar-Based 3D Human Tracking. Computer Vision and Image Understanding, 104, 178–189.

[12] N. Scherer-Negenborn and R. Schaefer (2010). Model Fitting with Sufficient Random Sample Coverage. International Journal of Computer Vision, 89, 120–128.

[13] G. Silveira and E. Malis (2010). Unified Direct Visual Tracking of Rigid and Deformable Surfaces Under Generic Illumination Changes in Grayscale and Color Images. International Journal of Computer Vision, 89, 84–105.

[14] A. Thomaz, H. Jacobsson, G. Kruijff, and D. Skocaj (2008). In Proceedings of the Interactive Robot Learning, Robotics: Science and Systems (RSS) 2008 Workshop, Zurich, Switzerland, June 28, 2008.

[15] A. Vedaldi and B. Fulkerson. (2010). VLFeat: An Open and Portable Library of Computer Vision Algorithms. www.vlfeat.org. Accessed 31 October 2010.

[16] M. Wyawahare, P. Patil, and H. Abhyankar (2009). Image Registration Techniques: An Overview. International Journal of Signal Processing, Image Processing and Pattern Recognition, 2(3), 11-27.

[17] B. Zitova and J. Flusser (2003). Image Registration Methods: A Survey. Image and Vision Computing, 21, 977–1000.

[18] Y. Sheikh, O. Javed, and T. Kanadel (2009). Background Subtraction for Freely Moving Cameras. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, Kyoto, Japan, September 29 2009-Oct. 2 2009, 1219–1225.

[19] D. O. Johnson and A. Agah (2011). A Novel Efficient Algorithm for Locating and Tracking Object Parts in Low Resolution Videos. Journal of Intelligent Systems, April 2011, 20(1):79-100, DOI: 10.1515/JISYS.2011.006

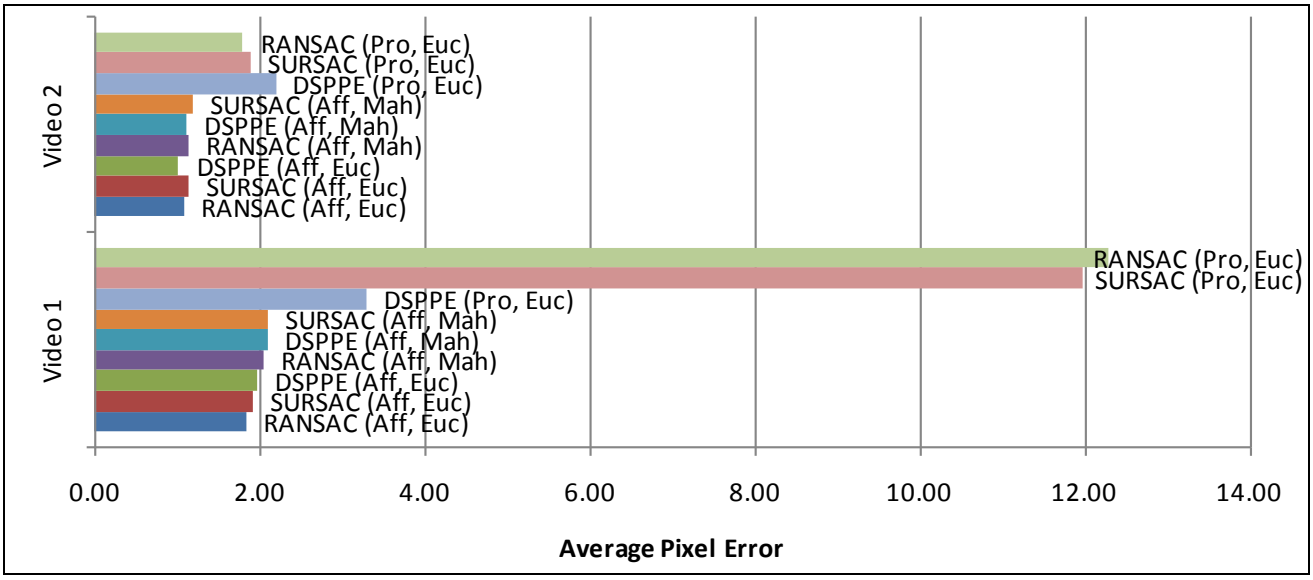


Figure 5. Comparison of various algorithm implementations by average pixel error. Each implementation used a different method of determining stationary points (i.e., DSPPE, RANSAC, or SURSAC), a different spatial transform (i.e., **Aff**ine or **Pro**jective), and a different fitness measure (i.e., **Euc**lidean or **Mah**alanobis).

Recognition of Marker-less Human Actions in Videos Using Hidden Markov Models

David O. Johnson[1] **and Arvin Agah**[2]
[1]Computer Science and Electrical Engineering, University of Missouri – Kansas City, Kansas City, MO, USA
[2]Electrical Engineering and Computer Science, University of Kansas, Lawrence, KS, USA

Abstract - *In this paper, we present a novel methodology for using Hidden Markov Models (HMM) to recognize marker-less human actions in videos. We use Scale Invariant Feature Transform (SIFT) keypoints to identify and track the trajectories of objects and major body parts in instructional videos. Then, we use a HMM to recognize the trajectories as a human action. This enables the use of readily available instructional videos from the Internet to train robots to perform tasks instead of programming them. The HMM we designed is able to recognize a marker-less human grasping an object in a video with a precision of 80%. The experiments also showed that the HMM can predict whether the human is reaching, is grasping, or has grasped the object with an accuracy of 75%.*

Keywords: Hidden Markov Model, Video Analysis, Robotics, Motion Analysis, Tracking Objects, Gesture Recognition

1 Introduction

Today there are no general-purpose service robots that can perform household activities. There are special purpose machines that can vacuum a room (e.g., the Roomba commercial robot [12]). However, only robots in laboratory environments can do so by manipulating a vacuum cleaner. It is envisioned that someday there will be general-purpose service robots that can perform these types of tasks. These general-purpose robots will need to be programmed to perform all kinds of human activities. The organizers of the Interactive Robot Learning Workshop, Robotics: Science and Systems 2008, held in Zurich, Switzerland in June, 2008 said that instead of programming these robots, we should teach them in the same way humans teach each other [24]. One cost-effective method of teaching is using instructional videos, as there are numerous instructional videos for teaching various types of skills. One challenge associated with using instructional videos is how does the vision system detect physical movements in a marker-less, environment, where the user does not wear special clothing or sensors? Inamura *et al*. used Hidden Markov Models to encode human joint trajectories into proto-symbols [10, 11]. In their experiments, they were able to show that a HRP-2W service robot could learn and demonstrate the proto-symbol by watching a human perform it. However, the joints of the human teacher were tracked using a wearable motion capturing system, which will not be likely in instructional videos. In this paper, we use Lowe's SIFT keypoints [19] to identify and track the trajectories of objects (e.g., scrub brush, pail of water) and major body parts (e.g., hand, head, body) in instructional videos. Then, we use a Hidden Markov Model to recognize the trajectories as a proto-symbol. This enables the use of readily available instructional videos from the Internet to teach robots to perform tasks instead of programming them.

This paper is organized into five sections. Section 2 discusses the related work. Section 3 describes the Hidden Markov Model we used and the measurements we used to verify it recognized the proto-symbols. Section 4 discusses the results of the experiments that we performed. Section 5 concludes the paper.

2 Related work

Various methods of video tracking and pose estimation have been proposed. Some of the more recent methods are discussed here. Deutscher and Reid used a modified particle filter, which they call annealed particle filtering to recover full articulated body motion from marker-less human motion [8]. John *et al*. used a hierarchical search algorithm to match human models of truncated cones with edges and silhouettes detected from multiple cameras to track the three-dimensional pose of a human [13]. Both of these methods rely on multiple cameras which would not work with single camera training videos. Ong *et al*. used a particle filter to match exemplars of the human skeleton with edges of limbs detected from an image to track the three-dimensional pose of a human across video frames from a single camera [23]. Chen and Schonfeld propose a method to track the object's motion and estimate its pose directly from two-dimensional image sequences using Lowe's Scale-Invariant Feature Transform (SIFT) [7, 19]. In our work, we want to use machine learning to recognize a sequence of n-dimensional vectors of x-y coordinates of inanimate object and body part

centroids as a specific proto-symbol. Video tracking and pose estimation, on the other hand, seek to extract features from the image (e.g., edges, silhouettes, SIFT keypoints) and perform a best match to known models of the human body in particular poses. Thus, the output is a sequence of body poses that can be replicated for animation, virtual reality, etc., or as an alternative to our method of representing poses as a sequence of body part centroids.

Robot Programming by Demonstration (RbD), also referred to as Learning by Imitation and Programming by Demonstration (PbD), is a non-verbal technique generally used to teach new motor skills to a robot. This approach works best for learning multi-dimensional and non-linear tasks. Billard *et al*. published an extensive review of Robot Programming by Demonstration [3], including a history of RbD, the many approaches to RbD, and a list of open issues. Klingspor *et al*. reviewed the state of the art in 1997 and identified two requirements for RbD [18]: The human teacher must be competent in the skill being demonstrated; and the robot acquiring the skill must have adequate sensors and degrees-of-freedom (DOF) to perform the skill. Nicolescu and Mataric taught a Pioneer 2DX robot to reach a set of colored columns in a specific sequence as demonstrated by a human [21, 22]. Calinon *et al*. demonstrated another example of learning by imitation [5]. The experiment starts with the human and the robot facing each other across a table. There is a red and a green dot on the table. The human reaches for each dot alternatively with the left or right arm. The robot then imitates the human. In another experiment, they taught a robot to move chess pieces by the user moving the robot's arm through the task [6]. Alissandrakis *et al*. addressed the RbD problem where the robot and the human may not share the same degrees-of-freedom, morphology, constraints, etc. [1]. In other work, Calinon and Billard explored the issue of recognizing, generalizing and reproducing arbitrary gestures [4].

Kim *et al*. showed how tasks could be decomposed into robot-independent, generic Molecular and Atomic Actions [17]. Kim *et al*. define Molecular Actions to be of such granularity as "find object" or "move to location". They usually involve a single object or location parameter, and their execution requires coordination of sensing, planning and action modules of the robotic system. Molecular Actions are similar to the proto-symbols used by Inamura *et al*. [10, 11]. The actual execution of Molecular Actions requires a sequence of one or more primitive Atomic Actions. Kim *et al*. define Atomic Actions to be a primitive operation for which completion can be verified with sensors (e.g., vision, sound). In their experiments, they showed that only 17 Molecular Actions were required to implement eight home-service robots tasks. The eight home-service robot tasks were bring me an object, throw an object, turn on a switch, move through a door, pour water, go up and down in the elevator, vacuum the floor, and guide a guest. They also showed that it took only 30 Atomic Actions to implement the 17 Molecular Actions. They implemented Molecular Actions in a real robot called IDRO (ICU DML Robot). The Digital Media Laboratory of KAIST (Korea Advanced Institute of Science and Technology) developed IDRO.

3 HMM for Molecular Actions

To validate our method of detecting Molecular Actions using the trajectories of the centroids of marker-less human body parts and the trajectories of the centroids of objects, we created a Hidden Markov Model to detect the "grasping an object" Molecular Action.

3.1 Hidden Markov Model

As illustrated in Fig. 1 we used a Hidden Markov Model (HMM) of the "grasping an object" Molecular Action with four states: No Action, Reaching, Grasping, and Grasped. The state of the HMM depends on the position and velocity of the Hand and the Object that the Hand is grasping. Table 1 explains the conditions for each state.

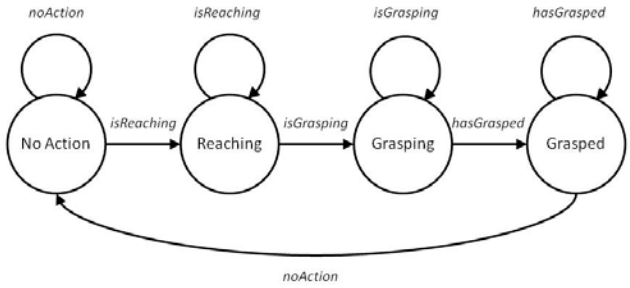


Figure 1. Hidden Markov Model of Grasping an Object Molecular Action

Table 1. Conditions for States of HMM of Grasping an Object Molecular Action

	Conditions for States			
	Hand		Object	
State	Velocity	Position	Velocity	Position
Reaching	moving toward Object	not close to Object	not moving	not close to Hand
Grasping	not moving	close to Object	not moving	close to Hand
Grasped	moving same as Object	close to Object	moving same as Hand	close to Hand

3.2 Implementing the Hidden Markov Model

Hidden Markov Models are defined by their state transition and emission matrices. The state transition matrix is an *n*-by-*n* matrix, where *n* is the number of states (e.g., $n = 4$ for "grasping an object"). Each element of the state transition matrix is the estimated probability of transition from state *i* to state *j*. The emission matrix is an *n*-by-*m* matrix, where *n* is the number of states and *m* is the number of possible emissions. Each element of the emission matrix is the estimated probability that emission *j* is emitted from state *i*. The state transition and emission matrices for our Molecular Action HMM were created from training data. For each of the training sets, we created a sequence of emissions, one for each frame of the video. In a perfect world, only one condition would be detected in each frame. However, in the real world, noise may cause more than one condition to be detected in each frame. To allow for noise, we defined eight emissions, one for each of the seven combinations of the three conditions (i.e., *isReaching*, *isGrasping*, and *hasGrasped*) and one for the emission where no condition is detected. The emission for each frame was determined by independently checking for the three conditions as shown below:

$$isReaching = C_1 \wedge C_2 \wedge C_3 \quad (1)$$

$$C_1 \leftrightarrow |tan^{-1}((p_{OY}-p_{HY})/(p_{OX}-p_{HX})) - tan^{-1}(v_{HY}/v_{HX})| < t_1 \quad (2)$$

$$C_2 \leftrightarrow (|v_{OX}| < t_2) \wedge (|v_{OY}| < t_2) \quad (3)$$

$$C_3 \leftrightarrow ((p_{OX} - p_{HX})^2 + (p_{OY} - p_{HY})^2)^{½} > t_3 \quad (4)$$

$$isGrasping = C_4 \wedge C_5 \wedge C_6 \quad (5)$$

$$C_4 \leftrightarrow ((p_{OX} - p_{HX})^2 + (p_{OY} - p_{HY})^2)^{½} < t_4 \quad (6)$$

$$C_5 \leftrightarrow (v_{HX}^2 + v_{HY}^2)^{½} < t_5 \quad (7)$$

$$C_6 \leftrightarrow (|v_{OX}| < t_6) \wedge (|v_{OY}| < t_6) \quad (8)$$

$$hasGrasped = C_7 \wedge C_8 \quad (9)$$

$$C_7 \leftrightarrow ((p_{OX} - p_{HX})^2 + (p_{OY} - p_{HY})^2)^{½} < t_7 \quad (10)$$

$$C_8 \leftrightarrow ((v_{OX} - v_{HX})^2 + (v_{OY} - v_{HY})^2)^{½} < t_8 \quad (11)$$

Where, p_{OX} and p_{OY} are the x and y-coordinates of the Object's position; p_{HX} and p_{HY} are the x and y-coordinates of the Hand's position; v_{OX} and v_{OY} are the x and y velocity of the Object; and v_{HX} and v_{HY} are the x and y velocity of the Hand. The velocity is calculated as follows:

$$v_j = (p_j - p_i) / (j - i) \quad (12)$$

Where v_j is the velocity of the Object or Hand in the x or y direction in the *jth* frame; p_i and p_j are the x or y coordinate of the Object's or Hand's position in the *ith* and *jth* frame, respectively; *j* is the number of the current frame; and *i* is the number of the previous frame where the Object or Hand was detected. Note that *i* and *j* may not be sequential because the Object or Hand may not be detected in all frames. The constants, t_1 through t_8, represent the tolerances allowed in detecting the conditions in Table 1, i.e., whether the Hand is moving toward the Object, the Hand and Object are close or not close, the Object is not moving, and the Hand and Object are moving with the same velocity.

The ground-truth state for each frame was determined by manually observing each frame of the video. The ground-truth state information and sequence of emissions derived from the training data were then used to estimate the probabilities for the emission matrix and the state transition matrix using the Viterbi algorithm [26]. We also tried training the HMM using the Baum-Welch algorithm [2], but got better and faster results using the Viterbi algorithm.

3.3 Detecting the Molecular Action

We can use the HMM to determine the probability that the human in the video is reaching, is grasping, or has grasped the object. In our application of robot programming by demonstration, it is not necessary to know what the human is doing to this granularity. It is only necessary to know 1) that the human performed the "grasped an object" Molecular Action, and 2) the order in which the human performed that Molecular Action with respect to the other Molecular Actions that the human performed in demonstrating the task.

To determine whether the HMM detected the Molecular Action or not, we looked at two methods. In the first method, we used the Viterbi algorithm to calculate the most likely path through the HMM given a test sequence of emissions. Then, we summarized the data by determining the most frequently occurring state in every 60 frames, which represented approximately two seconds of elapsed time. In the second method, we used the posterior state probabilities of the test sequence of emissions to determine the most probable state in every 60 frames. The posterior state probabilities are the conditional probabilities of being at state *k* in frame *i*, given the test sequence of emissions.

To determine whether the HMM detected the Molecular Action in the correct order, we next searched the summarized state data for transitions to the Grasped state, signaling that indeed the human had grasped the object. If a transition occurred any time during a ground-truth interval, which began when the human started reaching for the object and continued through the human grasping and

holding the object (i.e., grasped) until the human let go of the object, the transition was considered a true-positive detection of the Molecular Action. If the transition occurred outside this ground-truth interval, it was considered a false-positive. We then used the true-positive (TP) and false-positive (FP) counts to calculate the Precision (P) for the test data, as follows:

$$P = TP/(TP + FP) \quad (13)$$

3.4 Hidden Markov Model Accuracy (ACC)

Although not important to our application, we also measured the Accuracy (ACC) of the HMM in terms of how well it predicted the state of the HMM in each frame compared to the ground-truth state in each frame, as follows:

$$ACC = \sum(S_a = S_p) / \sum S_a \quad (14)$$

Where S_a is the actual, or ground-truth, state for the frame and S_p is the state predicted by the HMM for the frame. As discussed previously, the ground-truth state for each frame (S_a) was determined by manually observing each frame of the video. The predicted state (S_p) was calculated and reported in two different ways. In the first method, the Viterbi algorithm was used to compute the most likely sequence of states that the HMM would go through to generate the test data sequence of emissions. In the second method, we used the posterior state probabilities of the test sequence of emissions to determine the most probable state in each frame (S_p).

4 Experimental Results

To validate our algorithm, we performed k-fold cross-validation using five data sets of a Hand grasping an Object from two different low-resolution (240 pixels wide and 180 pixels high) instructional videos on cleaning golf clubs. We used Lowe's SIFT keypoints with a nearest neighbor object detection approach to track the centroids of 14 objects across the frames of each video [27]. Fig. 2 illustrates the centroids of the 14 objects detected in Video 1, and Fig. 3 illustrates them in Video 2.

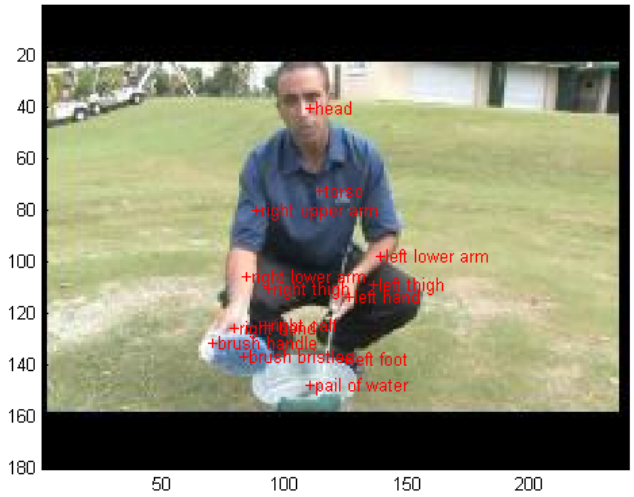


Figure 2. Centroids of the 14 objects detected in one frame of Video 1 [9]

Figure 3. Centroids of the 14 objects detected in one frame of Video 2 [9]

Table 2 shows which Video, Hand, Object, and frames were used for each data set.

Table 2. Data Sets Used in K-Fold Cross-Validation of HMM

Data Set	Video	Hand	Object	Frames
1	1	right hand	brush handle	1-2405
2	1	right hand	brush bristles	1-2405
3	2	right hand	brush	1-1964
4	2	right hand	golf club	1-1326
5	2	right hand	golf club	1327-1964

Table 3. Experimental Results

	Viterbi Algorithm					Max. Posterior State Probability				
Test Set	1	2	3	4	5	1	2	3	4	5
ACC	82%	82%	75%	83%	53%	81%	82%	74%	83%	53%
TP	1	1	1	0	1	1	1	1	0	1
FP	0	1	0	0	0	0	1	0	0	0

The experiments were implemented using the MATLAB Image Processing and Statistics (Hidden Markov Models) Toolboxes [20] and the VLFeat open source library of computer vision algorithms [25]. We conducted five experiments. In each experiment, we used four of the data sets as the training data to create the HMM and the other one to test it. The values for the condition tolerances were determined experimentally and were the same in all five experiments, with $t_1 = 0.88$, $t_2 = 10$, $t_3 = 46$, $t_4 = 41$, $t_5 = 2.9426$, $t_6 = 1.95$, $t_7 = 41$, and $t_8 = 7$. The results of each experiment are shown in Table 3.

The mean accuracy (ACC) was 75% using the Viterbi algorithm to determine S_p, and was also 75% using the maximum posterior state probability to determine S_p. The mean precision (P) was 80% using the Viterbi algorithm to detect the Molecular Action, and was also 80% using the maximum posterior state probability to detect it.

5 Conclusion

The HMM we designed is able to recognize a marker-less human grasping an object in a video with a precision of 80%. This enables the use of readily available instructional videos from the Internet to train robots to perform tasks instead of programming them. The experiments also showed that the HMM can predict whether the human is reaching, is grasping, or has grasped the object with an accuracy of 75%.

The presented results could be improved in a few ways. First, we assume the coordinate system remains the same across frames, which ignores small changes in the coordinate system between frames that are caused by the camera zooming in or out and panning. This could be corrected by using image registration. It could also be corrected by using the distance between the hand, lower arm, and upper arm as a scaling factor. This method is video independent because it relies on the geometry of the human body. Second the tolerances for detecting the conditions of the states were determined experimentally and the same values were used in all five experiments. Better results might be obtained by using the training data in each experiment to determine the tolerances using Gradient Descent or some other maximizing algorithm. Third, the noise in the trajectories of the human's hands and the objects could be filtered out using a Kalman filter [16] or a particle filter. Fourth, we could take advantage of the other predominant modality of an instructional video, speech, to improve the results. Johnson and Agah showed that using multiple modalities, dialog history, real world context, and dialog management can improve the task learning rate by as much as 20% over speech by itself [14, 15].

6 References

[1] Alissandrakis, A., Nehaniv, C., Dautenhahn, K.: Correspondence Mapping Induced State and Action Metrics for Robotic Imitation. IEEE Transactions on Systems, Man, and Cybernetics – Part B, 37(2), 299-307 (2007)

[2] Baum, L.E., Petrie, T., Soules, G., Weiss N.: A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. Ann. Math. Statist. 41(1), 164-171 (1970)

[3] Billard, A., Calinon, S., Dillmann, R., Schaal, S.: Robot Programming by Demonstration. In: Siciliano, B., Khatib, O. (eds.) Handbook of Robotics. Chapter 59. Springer (2008)

[4] Calinon, S., Billard, A.: Recognition and Reproduction of Gestures using a Probabilistic Framework combining PCA, ICA and HMM. In: 22nd International Conference on Machine Learning, Bonn, Germany, August 7-11, 2005, 105-112 (2005)

[5] Calinon, S., Guenter, F., Billard, A.: Goal-Directed Imitation in a Humanoid Robot 2005. In: IEEE International Conference on Robotics and Automation, Barcelona, Spain, April 18-22, 2005, 299-304 (2005)

[6] Calinon, S., Guenter, F., Billard, A.: On Learning the Statistical Representation of a Task and Generalizing it to Various Contexts. In: IEEE International Conference on Robotics and Automation, Orlando, FL, USA, May 15-19, 2006, 2978-2983 (2006).

[7] Chen, C., Schonfeld, D.: A Particle Filtering Framework for Joint Video Tracking and Pose Estimation. IEEE Transactions on Image Processing, 19(6), 1625-1634 (2010)

[8] Deutscher, J., Reid, I.: Articulated Body Motion Capture by Stochastic Search. International Journal of Computer Vision. 61(2), 185-205 (2005)

[9] eHow.com Website, www.ehow.com

[10] Inamura, T., Okadab, K., Tokutsub, S., Hataob, N., Inabab, M., Inoue, H.: HRP-2W: A Humanoid Platform for Research on Support Behavior in Daily Life Environments. In: 9th International Conference on Intelligent Autonomous Systems (IAS-9), Tokyo, Japan, March 7- 9, 2006, 57(2), 145-154 (2006)

[11] Inamura, T., Toshima, I., Nakamura, Y.: Acquiring Motion Elements for Bidirectional Computation of Motion Recognition and Generation. In: Siciliano B., Dario, P. (eds.) Experimental Robotics VIII, Springer-Verlag, Vol. 5, 372-381 (2003)

[12] iRobot Corporation Website, www.irobot.com

[13] John, V., Trucco, E., Ivekovic, S.: Markerless Human Articulated Tracking Using Hierarchical Particle Swarm Optimization. Image and Vision Computing, 28(11), 1530-1547 (2010)

[14] Johnson, D.O.: Human Robot Interaction through Semantic Integration of Multiple Modalities, Dialog Management, and Contexts. Ph.D. Dissertation, Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, KS, USA (2008)

[15] Johnson, D.O., Agah, A.: Human Robot Interaction through Semantic Integration of Multiple Modalities, Dialog Management, and Contexts. International Journal of Social Robotics, 1, September 2009, DOI: 10.1007/s12369-009-0028-0 (2009)

[16] Kalman, R.: A New Approach to Linear Filtering and Prediction Problems. Journal of Basic Engineering, 82(1), 35-45 (1960)

[17] Kim, H., Jung, Y., Hwang, Y.: Taxonomy of Atomic Actions for Home-Service Robots. Journal of Advanced Computational Intelligence and Intelligent Informatics, 9(2), 114-120 (2005)

[18] Klingspor, V., Demiris, J., Kaiser, M.: Human-Robot-Communication and Machine Learning. Universität Dortmund, Dortmund, Germany, University of Edinburgh, Edinburgh, UK, and Universität Karlsruhe, Karlsruhe, Germany (1997)

[19] Lowe, D.: Object Recognition from Local Scale-Invariant Features. In: International Conference on Computer Vision, 2, 1150-1157 (1999)

[20] The MATLAB Image Processing Toolbox 6.3. MathWorks. www.mathworks.com

[21] Nicolescu, M., Mataric, M.: Learning and Interacting in Human-Robot Domains. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 31(5), September 2001, 419-430 (2001)

[22] Nicolescu, M., Mataric, M.: Task Learning Through Imitation and Human-Robot Interaction. In Dautenhahn, K., Nehaniv, C. (eds.) Models and Mechanisms of Imitation and Social Learning in Robots, Humans and Animals, Cambridge University Press, Cambridge, UK (2005)

[23] Ong, E., Micilotta, A., Bowden, R., Hilton, A.: Viewpoint Invariant Exemplar-Based 3D Human Tracking. Computer Vision and Image Understanding, 104, 178–189 (2006)

[24] Thomaz, A., Jacobsson, H., Kruijff, G., Skocaj, D.: In: Interactive Robot Learning, Robotics: Science and Systems (RSS) 2008 Workshop, Zurich, Switzerland, June 28, 2008 (2008)

[25] Vedaldi, A., Fulkerson B.: VLFeat: An Open and Portable Library of Computer Vision Algorithms, www.vlfeat.org

[26] Viterbi, A.J.: Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. IEEE Transactions on Information Theory 13(2), 260--269 (1967)

[27] D. O. Johnson and A. Agah (2011). A Novel Efficient Algorithm for Locating and Tracking Object Parts in Low Resolution Videos. Journal of Intelligent Systems, April 2011, 20(1):79-100, DOI: 10.1515/JISYS.2011.006

Situations, Deduction, Plausibility

Peeter Lorents

NATO Cooperative Cyber Defence Centre of Excellence

Erika Matsak

Tallinn University, Estonia

Abstract. We consider time-dependent systems and their possible states or situations. In order to do this, we define suitable algebraic structures, including time. Situation descriptions are represented with the corresponding formulas and we look at inferences from formulas, where the inference may be partially incorrect. We define the measure of correctness and show how it can be used to compare the plausibility of the described situations.

Keywords. Time-dependent systems. Developments and deduction. Measure of correctness of inference. Plausibility.

1. Introduction

In this work we consider time-dependent systems. These are systems, where the sets of elements and the properties of and relations between elements are dependent on time (see Lorents, Matsak 2011). At any moment of time t the system M is in a certain state M(t), or in other words, there is a certain situation in the system (see Jakobson 2011). At some moment of time t', which comes after the moment t, the corresponding situation can be the same, or shorter - M(t)=M(t'). However, it can also change. In the latter case, where M(t)≠M(t'), we speak of an event (see, for example, Jakobson 2007, 2011).

It is often reasonable to handle the time-dependence of systems in such a way that the relation between time moments and possible states is not necessarily one-to-one (see Lorents, Matsak 2011; Lorents 2006). In such a case we need instruments that would help to ***compare*** the possible situations at the next moment. This is primarily necessary in order to find the most plausible future situation before making a decision on how to act.

There are several approaches to assess plausibility. For example, we can rely on the corresponding probabilities (see, for example, Polya 1954). Another approach is to use a suitably defined notion of possibility and the corresponding distribution functions that are related to so-called fuzzy sets, which have the values of real numbers from the range of [0,1] (see, for example, Zadeh 1978; D. Dubois, H. Prade, R. Sabbadin 2001; Jakobson 2011). The plausibility can also be handled in the framework of a certain multi-valent logic (see Sigarreta, Ruesga, Rodriguez 2007).

One option for assessing and comparing plausibility is to use the *principle of correlation of developments and deduction*, according to which the more logical an inference, where the starting point is the description DesM(t) of the current situation M(t) of the system M and the where the result is the description DesM(t') of a possible following situation M(t'), the more probable it is that this situation develops in the system (see Lorents 1998, 2006; Lorents, Matsak 2011).

At this point we need to explain how to understand the phrase "more logical". For this we use the notion of the correctness of inference steps (see, for example, Takeuti 1975) and we define the measure of correctness. By using the measure of the correctness of inference steps (which is represented by the rational number that we get by applying the corresponding procedure), we can compare single inference steps and also the reasoning based on these inference steps. Based on this, the corresponding descriptions, and relying on the principle of correlation of developments and deduction, we can compare the following possible situations, in order to make a decision about which one of them is more plausible.

According to the chosen approach, the key question is how to define the measure of correctness of inference steps, so that it

(1) would be in accordance with the notion of correctness of inference steps used in mathematical logic

(2) would allow to operate with comparable values, preferably numbers, and would have the corresponding algorithm for getting these values

(3) would allow to assess the correctness of an inference consisting of single inference steps

(4) would allow (based on the principle of correlation of developments and deduction) to assess, which inference process produces a description of a situation that will more plausibly take place compared to some other situation.

2. Systems and time

When dealing with systems in this work, we use the notion of an algebraic system (see, for example, Maltsev 1970; Grätzer 2008) and we look at such ordered pairs, where the first is the collection of sets of elements of

interest and where the second is the collection of the properties of or relations between the elements of interest.

Definition 1. A *system* is an ordered pair ⟨{X,Y,Z,…};{P,Q,R,…}⟩, where X, Y, Z, … are the *main sets of the system* (note that in a special case the system can have just one main set, for example X) and P, Q, R, ... are the *predicates of the system* (the properties or relations of the elements of the sets in question). The collection of predicates is called the *signature of the system*.

Example 1.1. Married people could conservatively be represented as the system ⟨{H};{♂,♀,Mar}⟩, where the only main set is the set H of humans and the signature consists of the properties "male", "female" and a binary marriage relation Mar. At the same time, we could look at the same people as the system ⟨{M, F};{Mar}⟩, which has two main sets: the set of men M and the set of women F; and which has only one binary relation, marriage relation Mar, in its signature.

According to the *Systematic principle*, things that can reasonably be represented as systems or belonging to systems, should be (see Lorents 1998, 2006). Therefore, we can also approach *time* from the systematic perspective:

Definition 2. *Time* is a system ⟨{T, D}; {Bef, Aft, Sim, Dur}⟩, where the main sets are the set of time moments T and the set of the lengths of time intervals D and where the signature consists of binary relations: the relation of being before Bef, the relation of being after Aft, the relation of being simultaneous Sim, and the ternary relation Dur, which relates any two observed time moments and the time interval, which includes all the time moments that occur between them.

Example 2.1. The budget year time

⟨{{01.01,02.01,…,30.12,31.12},{1,…,365}};{<,>,=,$-_{+1}$}⟩,

where $-_{+1}$ is a ternary relation, which relates two dates with the difference of the later date and the earlier date that is increased by one.

For example, if some work starts on the second of January and also finishes on the second of January (of the same year), then the number of days one should get paid for is $2-_{+1}2=2-2+1=1$ (NB! Not 2–2=0). If, however, the work begins on the second of January and ends on the twelfth of January, then one should get paid for $12-_{+1}2=12-2+1=11$ (NB! Not 12–2=10) days.

In order to define time-dependent systems (see Lorents, Matsak 2011) we must first identify the (concrete) system ∇=⟨{T,D};{Bef,Aft,Sim,Dur}⟩ that is our time and the classes CSets and CPred. The *elements* of class *CSets* are collections of sets or sets, where the elements are some other sets. The elements of class *CPred* are collections of predicates or sets, where the elements are predicates (let us recall that we have unary predicates as properties or subsets of some sets, and that we have k-order predicates (where k > 1) as relations between k elements or, in other words, the subset of the Cartesian product of some sets). Next we must identify the relations Set and Sig, which allow us to relate time moments with sets of elements from class CSets and signatures (or collections of the properties of elements and the relations between elements) from class CPred. Only now can we present the definition of a time-dependent system:

Definition 3 (Lorents, Matsak 2011)**.** An ordered triplet ⟨∇,Set,Sig⟩ is called a *time-dependent system*, if the following condition is fulfilled:

If, for some time moment t∈T, the collection of sets s∈CSets and the collection of predicates p∈CPred,

(1) Set(t,s) and (2) Sig(t,p),

then, without exceptions, all predicates from collection p must be the properties of or relations between the elements of the sets from collection s.

Let us agree that the *main sets of system* ⟨∇,Set,Sig⟩ *at time moment t can only be such sets* that are elements of collection s. Let us also agree that *only such predicates, which come from collection p, can be the predicates of system* ⟨∇, Set, Sig⟩ *at time moment t*. Finally, let us agree that *the possible state of the system* ⟨∇, Set, Sig⟩ *at time moment t* is the ordered pair ⟨s;p⟩, or a system where the collection of main sets is s (where Set(t,s)) and the signature (or collection of predicates) is p (where Sig(t,p)) and where, without exceptions, all predicates from collection p must be the properties of or relations between the elements of the sets in collection s). The collection of all possible states of the system is called the *development space of the system in time* ∇.

Example 3.1 Let us look at the system VF (Vooglaid family):

VF(2002)= ⟨**{**{Varro, Helena}**}**;{♂,♀,Mar}⟩ (Mar represents the relation "are married")

VF(2003)=⟨**{**{Varro,Helena},{Iida}**}**;{♂,♀,Mar,Par}⟩ (Par represents the relation "is a parent of"; the set {Varro,Helena} is the set of spouses in year 2003 and {Iida} is the set of children in year 2003)

VF(2004)=⟨**{**{Varro,Helena},{Iida}**}**;{♂,♀,Mar,Par}⟩

VF(2005)=⟨**{**{Varro,Helena},{Iida}**}**;{♂,♀,Mar,Par}⟩

VF(2006)=⟨**{**{Varro,Helena},{Iida,August}**}**;
{♂,♀,Mar,Par,BoS}⟩ (BoS represents the relation "is a Brother or Sister of")

… … …

VF(2008)=⟨**{{**{Varro,Helena},{Iida,August}**}**;
{♂,♀,Mar,Par,BoS}⟩

VF(2009)=⟨**{{**{Varro,Helena},{Iida,August,Benita}**}**;
{♂,♀,Mar,Par,BoS}⟩.

In here CSets=**{{**{Varro,Helena}**},{**{Varro,Helena},{Iida}**}**,
{{Varro,Helena},{Iida,August}**}**,
{{Varro,Helena},{Iida,August,Benita}**}}** and

CPred={{Mar},{Mar,Par}, {Mar,Par,BoS}}.

Set ⊆ {02,03, …,09}×
×**{{**{Varro,Helena}**},{**{Varro,Helena},{Iida}**},**

{{Varro,Helena},{Iida,August}**},{**{Varro,Helena}**,**
{Iida,August,Benita}**}}**

Set=**{**⟨02,**{{**Va,He}**}**⟩,⟨03,**{{**Va,He},{Ii}**}**⟩,
⟨04,**{{**Va,He},{Ii}**}**⟩,⟨05,**{{**Va,He},{Ii}**}**⟩,
⟨06,**{{**Va,He},{Ii,Au}**}**⟩,⟨07,**{{**Va,He},{Ii,Au}**}**⟩,
⟨08,**{{**Va,He},{Ii,Au}**}**⟩,⟨09,**{{**Va,He}**,** {Ii,Au,Be}**}}**⟩**}**

Sig ⊆ {02,03, …, 09}×**{{**♂,♀,Mar}**,** {♂,♀,Mar,Par}**,**
{♂,♀,Mar,Par,BoS}**}**

Sig=**{**⟨02,{♂,♀,Mar}⟩,⟨03,{♂,♀,Mar,Par}⟩,
⟨04,{♂,♀,Mar,Par}⟩,⟨05,{♂,♀,Mar,Par}⟩,
⟨06,{♂,♀,Mar,Par,BoS}⟩,⟨07,{♂,♀,Mar,Par,BoS}⟩,
⟨08,{♂,♀,Mar,Par,BoS}⟩, ⟨09,{♂,♀,Mar,Par,BoS}⟩**}**.

3. Describing situations

By studying the descriptions of time-dependent systems at one or another time moment, we can see that in a large part of the cases the descriptions consist of arguments in natural language text (for example, some elements have these properties but lack those properties; some elements are related somehow, but others are not; if some elements have such properties, then they are not related in that way, etc.). If we transform texts like this (for example, by using the DST dialogue system (see Matsak 2005; Matsak 2010; Lorents, Matsak 2011) we get certain predicate calculus formulas.

Example 3.1.1. We can describe the situation of family Vooglaid in 2007 with, for example, the following formulas: ♂(Va), ♂(Au), ♀(He), ♀(Ii), Mar(Va,He), Par(Va,Ii), Par(Va,Au), Par(He,Ii), Par(He,Au), BoS(Ii,Au).

For "aliens" who are not very familiar with Earthly affairs, we can add the following: $\forall h(♂(h)\vee♀(h))$, $\forall h(♂(h)\Leftrightarrow\neg♀(h))$, $\forall h_1h_2(Mar(h_1,h_2)\supset(\neg(♂(h_1)\&♂(h_2))\vee\neg(♀(h_1)\&♀(h_2))$, $\forall h_1h_2(Mar(h_1,h_2)\supset\neg BoS(h_1,h_2))$, $\neg(\exists\forall h_1h_2)(BoS(h_1,h_2)\&Mar(h_1,h_2))$, etc.

By assigning to individual and predicate symbols suitable meanings drawn from the main sets and signature of the system, we get knowledge related to that specific system (see Lorents 2009). This knowledge is necessary both for handling the current state of the system, as well as for getting knowledge about the states of the system or situations at some later time moments.

One option for such forecasting is to rely on deduction, which people often do. However, they do not always use logically correct inference steps. On the other hand, we must admit that people often perform quite successfully even without "iron logic" (see, for example, Rescher 1976). In other words (NB!) *people can be relatively successful even if not all the inference steps in a deduction process are logically correct.* This brings us to the **main problems of this work:**

- how to assess the correctness of given inference steps or inferences that contain them? This problem is related to the observation that "good" decisions, which led to "good results" are often based on "better logic" (compared to others). Therefore, if it is not possible to build the decision process to full extent on logically correct inference steps, then what is our expectation that a situation matching the description that was generated in such a way (using somewhat broken logic) actually occurs?
- if and how can we ***compare*** the plausibility of the arrival of the described future situations, based on the measure of correctness of the inference steps that make up the deduction process?

The first step to solve these problems is to find a suitable definition for the measure of correctness of inference steps.

4. Measure of correctness of inference steps

In mathematical logic, we call an inference step *correct* if the following condition is met: for **any** interpretation, **if** for some interpretation φ **all** the direct premises of the observed inference steps are correct, **then** in the same interpretation φ the direct conclusion of the observed inference step must also be correct (see, for example, Takeuti 1975).

At this point we turn our attention to the phrase "for some interpretation φ **all** the direct premises of the observed inference steps are correct". It is here where we exclude all interpretations, where all the direct premises are not correct at the same time (in the framework of the given interpretation), from our process of defining the measure of correctness. From the remaining interpretations we choose such interpretations that also have the correct direct conclusion. When we compare the collection of chosen interpretations to the previous collection, we find the corresponding proportion, which

we take to be the measure of the correctness of the given inference step.

Definition 4. Let us observe an inference step, which has the formulas X, Y, …, Z as direct premises and the formula W as the direct conclusion. Let us look at interpretations, where the meanings of the individuals and predicates present in formulas come from a system with finite elements and signature. Let us look at the set of interpretations Pres={φ| φX=1&φY=1&…&φZ=1} and Cons={φ| φX=1&φY=1&…&φZ=1&φW=1}. The *measure of correctness of the inference step S* is the number cor(S), where cor(S)=E(Cons):E(Pres), if E(Pres)>0 and cor(S)=0, if E(Pres)=0, where E(H) represents the number of elements of a finite set H.

Example 4.1. From the reasoning of a two and a half year old child we find out that if it is cold outside then one cannot be naked outside, but if it is warm then one can (see http://childes.psy.cmu.edu/browser/index.php?url=Other/Estonian/Vija/20612.cha). This is an inference step, where the only direct premise is the formula X⊃¬Y and the direct conclusion is the formula ¬X⊃Y. It is not difficult to find that E(Cons):E(Pres)=2:3. Therefore, the measure of correctness of this inference step is 2:3.

Example 4.2. Why does our country have so many enemies, asks a youth from a known politician. Well, says the politician, on the one hand, of course, it is because if a country is wealthy and strong, many will want to be friends (or at least not enemies). On the other hand, wealth and strength is not liked by competitors and therefore all other countries are our enemies. In this case the wily politician seems to use the inference step, where the direct premises are the formulas X&Y⊃Z and X&Y⊃¬Z and the direct conclusion is ¬Z. The measure of correctness of such an inference step is 1:2.

Note. The fact that an inference step is performed "according to rules" does not automatically guarantee that the inference step is correct!

Example 4.3. Let us consider a rule that belongs to sequential predicate calculus inference rules – *introducing disjunction into the antecedent* (see Gentzen 1936, Takeuti 1975). Let us now implement an inference step, where the direct premises are sequences X→X&¬X and ¬X→X&¬X and the direct conclusion is the sequence X∨¬X→X&¬X. The measure of correctness of such an inference step is 0.

Let us now consider separately certain inference steps, which "introduce" quantifiers (that is, inference steps where the direct premise consists of formulas $F(b_1)$, $F(b_2)$, …, $F(b_m)$ and the direct conclusion is the formula ∀xF(x) or the formula ∃xF(x), where b_1, b_2, …, b_m are some elements of the finite system used for interpretation, m≤n and n is the number elements of the finite system used for interpretation). Based "directly" on the definition above (4), the measure of correctness in such inference steps is somewhat trivial: if there exists an element e in the system, so that φF(e)=0, then the measure of correctness of the step that introduces the universal quantifier is 0 in every case. On the other hand, if the system contains an element d, so that φF(d)=1, then the measure of correctness of the step that introduces the existential quantifier is 1 in every case.

However, people often use the above described inference steps in a somewhat different way: *the generalization is considered correct based on a finite number of suitable examples*. One possible explanation for this is people's weird (but ancient) belief that a collection of positive examples supporting some argument is enough to consider the argument proven for *all* cases. In such an approach, the measure of correctness of the step that introduces a universal quantifier is not at all important. In such cases some "measure of conviction" is much more important for people. There are undoubtedly several options for also defining this measure:

Let us observe an inference step, where the premises are the formulas $F(b_1)$, $F(b_2)$, …, $F(b_m)$ and the direct conclusion is the formula ∀xF(x), where b_1, b_2, …, b_m are such elements of the finite system used for interpretation, that $\varphi F(b_1)=1$, $\varphi F(b_2)=1$, …, $\varphi F(b_m)=1$ where m≤n and n is the number of elements of the finite system used for interpretation. In such a case we can take as the *measure of conviction*, for example:

(Definition 4A). The number m:n (which shows, what proportion of the full collection is represented by the "positive examples" that are offered as proof).

(Definition 4B). The representation of the selection b_1, b_2, …, b_m in the n-element collection (see Gliner J. A., Morgan G. A., Leech N. L. 2009).

Note. It should be safe to agree with the argument: more correct is more convincing. The inverted argument: more convincing is more correct, is not very convincing.

Example 4.AB. (see Matsak 2010) A child (5 years and 9 months old) reasons why flowers do not talk. The child uses the premise that she has never heard the voice of a flower in order to reach this conclusion. By transforming the text we get to an inference step, where the premise consists of a finite number of formulas $\neg Speak(f_1)$, $\neg Speak(f_2)$, …, $\neg Speak(f_m)$ (where f_1, f_2, …, f_m correspond to the flowers that the child has experience with), and the conclusion is the formula ∀f(¬Speak(f)). Convincing, right?

5. The measure of correctness of inference

The notion of the measure of correctness of inference is similar to the measure of correctness of inference steps.

In order to define it, we use the definition of a *tree shaped inference*:

Definition 5.

- Every inference step

$$\frac{X_1\ X_2\ \ldots\ X_k}{Y}$$

is an inference, where the premises are the direct premises $X_1, X_2, \ldots, X_k$ of this inference step and the conclusion is the direct conclusion Y of this inference step.

- Let us have inferences $D_1, D_2, \ldots, D_n$, where the premises are correspondingly $A_{11}, A_{12}, \ldots, A_{1m(1)}$, $A_{21}, A_{22}, \ldots, A_{2m(2)}, \ldots \ldots, A_{n1}, A_{n2}, \ldots, A_{nm(n)}$ and the conclusions are correspondingly $B_1, B_2, \ldots, B_n$. In addition, let us have such an inference step, where the direct premises are $B_1, B_2, \ldots, B_n$ and the direct conclusion is C. The inference of such a case is

$$\frac{D_1\ D_2\ \ldots\ D_n}{C}$$

where the premises are $A_{11}, A_{12}, \ldots, A_{1m(1)}, A_{21}, A_{22}, \ldots, A_{2m(2)}, \ldots \ldots, A_{n1}, A_{n2}, \ldots, A_{nm(n)}$ and where the conclusion is C.

Definition 6. Let us look at an inference D, which consists of inference steps $S_1, \ldots, S_u$. Let the corresponding measures of correctness of the inference steps be $c_1, \ldots, c_u$. The *internal measure of correctness* of the inference is the number $Incor(D)=\min\{c_1, \ldots, c_u\}$.

Definition 7. Let us look at an inference D, where the premises are $A_{11}, A_{12}, \ldots, A_{1m(1)}, A_{21}, A_{22}, \ldots, A_{2m(2)}, \ldots \ldots, A_{n1}, A_{n2}, \ldots, A_{nm(n)}$ and where the conclusion is C. Let us consider the interpretations, where the meanings of the individuals and predicates in the formulas come from a system with finite elements and signature. Let us look at the set of interpretations $Pres(D)=\{\varphi \mid \varphi A_{11}=1\&\varphi A_{12}=1\&\ldots\ldots\&\varphi A_{nm(n)}=1\}$, and the set $Cons(D)=\{\varphi \mid \varphi A_{11}=1\&\varphi A_{12}=1\&\ldots\&\varphi A_{nm(n)}=1\&\varphi C=1\}$. The *measure of external correctness* of the inference D is the number Excor(D), where $Excor(D)=E(Cons(D)):E(Pres(D))$, if $E(Pres(D))>0$ and $Excor(D)=0$, if $E(Pres(D))=0$, where E(H) represents the number of elements of the finite set H.

This raises the next **problem**: how are the external and internal measures of correctness of the inference related?

If we follow the ancient folk wisdom that the strength of the chain is determined by its weakest link (or in other words, the chain can be no stronger than the individual links), then the inequality $Excor(D)\leq Incor(D)$ should apply. If, however, we follow the idea that one can assemble quite reliable systems (for example, buildings, machinery, software etc.) from components which include some unreliable ones, then the inequality $Excor(D)\geq Incor(D)$ could apply.

Unfortunately, it turns out that neither case is true!

Theorem. (I) There exist inferences, where the external measure of correctness is strictly larger than the internal measure of correctness. (II) There exist inferences, where the external measure of correctness is strictly smaller than the internal measure of correctness.

Idea of the proof. To construct the necessary inferences, it is enough if in each so-called thread (see, for example, Takeuti 1975) the inference steps consist of such formulas, where the corresponding measures of correctness of the steps form a sequence of numbers, so that

- for "option (I)": no number is greater than E(Cons(D)):E(Pres(D)) and some number(s) are smaller than E(Cons(D)):E(Pres(D))
- for "option (I)": no number is smaller than E(Cons(D)):E(Pres(D)) and some number(s) are greater than E(Cons(D)):E(Pres(D)).

Example 7.1. Let us consider the tree shaped inference constructed in predicate calculus, which includes some incorrect inference steps:

$$\frac{\dfrac{X\rightarrow Y\vee X}{X\rightarrow Y\&X}\qquad\dfrac{Y\rightarrow Y\vee X}{Y\rightarrow Y\&X}}{X\&Y\rightarrow Y\&X}$$

The measure of correctness of the two inference steps in the top is 3:4. The measure of correctness of the bottom step is 1. The external measure of correctness of the entire inference is 1, but the internal measure of correctness is 3:4.

Example 7.2.

Let us consider the tree shaped inference constructed in predicate calculus, which includes some incorrect inference steps:

$$\frac{\dfrac{\rightarrow\neg X\vee Y}{\rightarrow Y}\qquad\dfrac{X\rightarrow Y}{\rightarrow Y}}{\rightarrow X\&Y}$$

The measure of correctness of the two inference steps in the top is 2:3. The measure of correctness of the bottom step is 1:2. The external measure of correctness of the entire inference is 1:3, but the internal measure of correctness is 1:2.

6. Inferences and comparing plausibility

We start with the approach of G. Jakobson, since it seems to best fit with the systematic approach that is based on

the systematic principle (see section 2) and used by the authors. Therefore (see Jakobson 2011): *We define plausible future situations as situations that in some dynamic system with some degree of likelihood could happen at some time moment in future.*

The development of time-dependent systems, or in Jakobson's terminology - dynamic systems, in real macro world seems to happen according to the following principles (Lorents, Matsak 2011; Lorents 2006; Lorents 1998):

- *The principle of diversity of development opportunities* (A time-dependent system can have multiple different possible states for the next time moment)
- *The principle of being in one state* (Each time-dependent system is in each time moment t in exactly one state of all of its possible states)
- *The principle of no predestination* (The system arriving to a specific state in time moment t is based on chance – it is an event with some probability)
- *The principle of the correlation of developments and deduction* (A system is more probable to transit from the current situation to a new situation, when that situation's description is more correctly inferable from the current situation's description).

Presuming that greater probability corresponds with greater likelihood, we can use the above described to formulate the following principle:

- For the possible future situations of a time-dependent system, the degree of likelihood that is based on ***what is known*** is greater for a situation, where the description has a ***known*** greater measure of correctness in the inference that follows from the description of the current situation.

Note. The phrases "what is known" and "known" are very important in this part.

Indeed, in order to make decisions for a situation that may develop, we can only use our knowledge (of the system and situation in question) and what is provided by chance. Knowledge, however, is represented by certain atomic formulas (see Lorents 2009), which form the formulas (or arguments written down in a "very strict form") necessary to describe the system and the situations. When comparing the reasoning for the appearance of several possible future situations, we can still only use what we know and what we have. We just do not have anything else at our disposal. If we must decide which situation will occur in the situation, then we must very clearly identify the ***known*** descriptions of future situations and the inferences (or reasoning) that lead to those descriptions (since we are talking about the future!). Therefore, we can only compare ***known*** inferences, in order to make the decision on their measure of correctness. Thereat (NB!) it is possible that someone else who knows something else than we do at the moment, will provide a completely different assessment of the situation. There is no way around it.

Important question: Which measure of correctness did we just discuss?

Answer: Definitely only one of two – either internal or external measure of correctness, not two together or combined in some way!

Recommendation: It looks like it is "technically" easier to use the internal measure of correctness for plausibility. This is because the correctness of single steps is easier to assess.

Presuming that greater probability and likelihood corresponds to greater plausibility, we can formulate the most important principle of this work:

- For two possible future situations of a time-dependent system, the plausibility that is based on ***what is known*** is greater for such a situation, where the description follows the inference (that is based on the description of the current system) with the ***known*** higher measure of correctness, or in a more formal representation, if:

(1) at the time moment t', which follows the current time moment t, it is possible that the system M transits into situation $M_1(t')$ or into situation $M_2(t')$, and

(2) the reasoning used is based on the description of the current situation DesM(t), and

(3) it turns out that the reasoning $(DesM(t) \vdash_1 DesM_1(t'))$ behind the description $DesM_1(t')$ is "more correct" compared to the reasoning $(DesM(t) \vdash_2 DesM_1(t'))$ behind the description $DesM_2(t')$,

then ***for situations related with these specific descriptions and reasoning***, it is more plausible that the situation M(t) transits into situation $M_1(t')$ and less plausible that it transits into situation $M_2(t')$, or shorter -

$[Incor(DesM(t) \vdash_2 DesM_2(t')) < Incor(DesM(t) \vdash_1 DesM_1(t'))] \supset [M_2(t') <_{plaus} M_1(t'))]$.

Conclusion. The plausibility (based on what is known) of possible situations is greatest for the ones, where the description (consisting of corresponding formulas) has a logically impeccable (known!) inference.

Summary. In this work we considered how time-dependent systems transition to various possible situations and we explored which one of these transitions

is more plausible. In order to do this we defined the necessary algebraic systems, including time as a system and the time-dependence of systems. It is possible to use formulas constructed from knowledge for describing the states of systems or situations. This way we can see if the description of one situation can be inferred from the description of another situation. We also considered a situation where not all inference steps are logically correct. By defining the measure of correctness we can compare the plausibility of possible future situations. We do this by comparing the measures of correctness of the inferences constructed to define and reason the corresponding descriptions.

References

Dubois D., Prade H., Sabbadin R. (2001). *Decision-theoretic foundations of qualitative possibility theory* . European Journal of Operational Research 128 (2001) 459-478.

Gentzen G. (1936). *Untersuchungen über das logische Schließen. I, II.* Mathematische Zeitrschrift, B.39. 176 – 210, 405 – 443.

Gliner J. A., Morgan G. A., Leech N. L. (2009). *Research methods in applied settings : an integrated approach to design and analysis.* New York; London : Routledge, 2009 2nd ed

Grätzer G. (2008). *Universal Algebra*. Second Edition. Springer.

Jakobson G. (2011). *Extending Situation Modeling with Inference of Plausible Future Cyber Situations.* The 1st IEEE Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA 2011). Miami Beach, FL.

Lorents, P. (1998) *Süsteemse käsitluse alused. Riigikaitse ja julgeoleku põhiküsimused*. Tallinn: Eesti Riigikaitse Akadeemia kirjastus. [Foundations of the Systematic Approach. Main Problems of National Defence and Security]

Lorents P. (2006). *Süsteemide maailm* (*The World of Systems*). Tartu: Tartu Ülikooli Kirjastus.

Lorents P. (2009). *Knowledge and Logic.* Proceedings of the International Conference on Artificial Intelligence. IC – AI'2009. Volume II, p. 568 – 570. Las Vegas, Nevada, USA. 2009

Lorents P., Matsak E. (2011). *Applying time-dependent algebraic systems for describing situations.* The 1st IEEE Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA 2011). Miami Beach, FL

Maltsev A. I. (Мальцев А. И.) (1970). *Алгебраические системы* (*Algebraic systems*). Moscow. Наука.

Matsak E. (2005). *Dialogue System for Extracting Logic Constructions in Natural Language Texts.* The International Conference on Artificial Intelligence, Las Vegas, USA.

Matsak E. (2010). *Discovering Logical Constructs from Estonian Children Language.* Germany: Lambert Academic Publishing.

Polya G. (1954). *Patterns of Plausible Inference*. Princeton, NJ.

Rescher N. (1976). „*Plausible Reasoning: An Introduction to the Theory and Practice of Plausibilistic Inference*". Assen/Amsterdam: Van Gorcum, 1976 pp. Xiii+124.

Sigarreta J. M., Ruesga P., Rodriguez M. (2007). *On Mathematical Foundations of the Plausibility Theory.* International Mathematical Forum, 2, 2007, no. 27, 1319 – 1328:

Takeuti G. (1975). *Proof Theory*. North-Holland Publishing Company. Amstardam, London.

Zadeh L. (1978). *Fuzzy Sets as the Basis for a Theory of Possibility*, Fuzzy Sets and Systems 1:3-28, 1978. (Reprinted in Fuzzy Sets and Systems 100 (Supplement): 9-34, 1999.)

Social Features Discovery from Cellphone Contextual Data by Semantic Location Classification

Werner Creixell, *Member, IEEE*, Tomás Arredondo, *Member, IEEE*,
Sebastián Contreras, Patricio Olivares and Wladimir Ormazábal

***Abstract*— How much can the places we visit in our daily life reveal about us? Is it possible to successfully assert a subject's location and affiliation via the data obtained from cell phones? In the present research we present a cascade binary classifier framework to infer a subject's location and affiliation. In the first stage, the user location is obtained by combining clustering and a location classifier. In the second stage, the contextual spatio-temporal data is used to discriminate the user affiliation. This framework is tested using a dataset of actual cellphone usage and bluetooth encounters. Three binary classifiers are used in each stage to validate the proposed method. Encouraging results are obtained showing a high classification success rate across the whole user base.**

I. Introduction

At the end of 2009 the estimated number of mobile phone users reached 4.6 billions [1]. About 67% of the world population is carrying communication devices with computing capabilities. The mobile phone has become the ubiquitous computer, with users growing in greatest numbers than any other communication or computational system. The implications and opportunities of this phenomenon are enormous. A mobile phone operates in an intimate relation with the user: it goes wherever the user goes, travels at the same speed, experiments the same temperature and air pollution, meets the same people and can sense the same equipment as the user. The mobile phone is thus becoming an ideal tool for learning more about human behavior [2].

There have been several projects that have looked to take advantage of the intrinsic co-location characteristic of owners and their cellular telephones. The firsts approaches to study human behavior tried to learn more about mobility patterns. González et. al. [4], studied the trajectory of 100,000 anonymized mobile users for a period of six months. Their analysis showed that after correcting for the differences in travel distance and their anisotropy, the individual travel patterns can be modeled by a single spatial probability function.

Extracting mobility data from CDR (Call Data Records), provided from operators, has the shortcoming that its precision is determined by the cellular tower coverage area, which highly varies from less than 1 km^2 in highly populated areas to more than 4 km^2 in others. The CDR data is also not periodically taken, normally CDR can give an estimated location only when the phone is used in a call. Eagle et. al. [5] solved this problem by logging the cellular tower ID data in the phone itself. Unsupervised clustering techniques were used to infer relevant location, then these locations were used as the states in a Dynamic Bayesian Network for mobility prediction with over 90% of accuracy.

In the work of Ashbrok et. al.. [6], the location data from a single user carrying a GPS receiver was logged during four months. Using the resultant data they proposed a method based on clustering to identify meaningful locations among the collected data, they also proposed a Markovian model for the user "transitions" between the locations. The model was then used to perform predictions, the system was also tested in a multiuser environment.

The project presented in [8] conducted an experiment by collecting various data types from cell phones. This data set was used to estimate the user's position, by cell tower ID patterns, user's daily routine and user's social networking. The Reality Mining project data was used in [9] to further investigate the structure in the data set. They extract the behavioral structure using eigenvectors which are named by them eigenbehaviors, where an individual behavior, in one day, can be represented by the weighted sum of its primary eingenbehaviors. This approach uses the first half of the day to calculate the weights and the user behavior in the rest of the day is estimated with a 76% of accuracy. Other work that uses this data set is found in [10], in this work the authors seek to discover the daily routine of users from the data set. To do this, a framework was constructed from two Hierarchical Bayesian models based on routine as given by location: Latent Dirichlet. al.location and Author Topic Model.

In this current research, we propose a cascade classifier framework to determine a particular user affiliation. The first stage of the cascade classifier is to determine meaningful locations, these locations are used as input to the second stage to determine the user affiliation. The framework is tested using the Reality Mining data set [7] with the locations *home, work* and *else*. In both stages we contrast three different ma-

Werner Creixell is with the Department of Electronic Engineering, Universidad Técnica Federico Santa María, Valparaíso, Chile. (phone: 56-32-2654758; fax: 56-32-2797469; email: werner.creixell@usm.cl). He is also visiting researcher with CSIS at The University of Tokyo, Japan.

Tomás Arredondo is with the Department of Electronic Engineering, Universidad Técnica Federico Santa María, Valparaíso, Chile. (phone: 56-32-2654212; fax: 56-32-2797469; email: tomas.arredondo@usm.cl).

Sebastián Contreras is with the Department of Electronic Engineering, Universidad Técnica Federico Santa María, Valparaíso, Chile. (phone: 56-32-2654758; fax: 56-32-2797469; email: cba.contrerasm@gmail.com).

Patricio Olivares is with the Department of Electronic Engineering, Universidad Técnica Federico Santa María, Valparaíso, Chile. (phone: 56-32-2654758; fax: 56-32-2797469; email: paolivaresr@gmail.com) .

Wladimir Ormazábal is with the Department of Electronic Engineering, Universidad Técnica Federico Santa María, Valparaíso, Chile. (phone: 56-32-2654212; fax: 56-32-2797469; email: wladimir.ormazabal@gmail.com).

chine learning based classification methods: Support Vector Machines (SVMs)[11], Naive Bayes (NB)[12] and Neural Networks (NNs)[13]. To the best of our knowledge, this is the first time that both affiliation and location information have been successfully extracted from such data.

This work is organized as follows: Section II reviews previous approaches toward extracting meaningful social information from data. Section III provides a description of the proposed framework. In Section IV we present the data models being used, the experiments being performed and the obtained results. Finally, Section V elaborates the conclusions from this study and some proposals for future work.

II. Previous Work

In this section we review previous research on identifying different meaningful social human behaviors using a variety of data sources, methods and objectives. These different efforts can be grouped according to their primary objective. Some of the objectives previously pursued by other authors include: activity discovery, location discovery, inferring user transportation routines and user affiliation identification.

A. Activity Discovery

The work of Huynh et. al. [18] introduces an approach to model and recognize daily routines such as commuting or office work from wearable sensors. Toward this they utilize probabilistic topic models to recognize routines as a composition of activity patterns (both annotated and from sensor data directly) in an unsupervised fashion.

Choudhury and Pentland [19], extract the social structures within human groups based on wearable sensors. A portable sensor is used to build models of face-to-face interactions within groups, detect when members are in face-to-face proximity and also when they are in conversations. It specifically detects: information about nearby people (IR sensor), speech information (microphone) and motion information (accelerometer). The continuous sensing of these variables permits the determination of the structure of communication networks.

B. Location Discovery

Meneses and Moreira [20] present a model that recognizes frequent places. It uses GSM cellID information obtained from cellular handsets and without any network topology information or user intervention. A familiarity index is estimated for each identified place which can be used by context-aware applications.

In the work of Hightower et. al. [21], GSM and WiFi signal fingerprint are used to determine when a user returns to a particular location. The salient locations are identified by a customized algorithm. In a similar fashion to [20], this approach is useful for finding frequent visited places, but can not provide semantic information without user intervention.

Geographic data clustering is used in [22] to identify meaningful places. Extended density based clustering is used to identify places from data. The obtained results are compared against the general purpose K-means clustering algorithm. In [23] a non-parametric Bayesian approach is used to identify salient locations from data. This approach is also contrasted against the K-means clustering algorithm.

C. User Transportation Routines Discovery

In the context of learning transportation routines, Liao et. al. [3] propose a hierarchical Markov chain based model of a user from data obtained by a portable GPS unit. They utilize a Bayesian network to represent the model and a particle filter to perform inference in an unsupervised manner that combines learning user location and transportation mode.

Calabrese et. al. [24], perform an analysis of crowd mobility during special social events. One million cell-phone traces are analyzed and their home locations are associated with these events. They show that the home location of people attending events is strongly correlated to the type of event. This has implications in city management, with implications in event management and congestion mitigation.

D. Inferring Individual Affiliation

To the best of our knowledge, there is not much previous work reported on the subject of user affiliation identification. One exception is the paper of Eagle and Pentland [9] which proposes a technique based on principal component analysis (PCA) that defines a behavior space. It suggests that the distance between an individual's representation in the behavior space to the principal component of one affiliation group can be used to infer the user's affiliation. Unfortunately, that is not the main objective of the paper and consequently there is not much evidence about the effectiveness of the method that can be contrasted with our proposal.

E. Classifiers

In our approach, we utilize hierarchical cascade classifiers as a means to infer user affiliation from an incomplete dataset. What follows is a review of the available results on using such classifiers and datasets.

1) Cascade Classifiers: In the work of Viola and Jones [15], a cascade AdaBoost [16] based classifier network has been used in computer vision in order to speed up recognition of faces over a complete dataset up to real time speeds with excellent detection rates.

2) Hierarchical Cascade Classifiers: In the work of Lu and Drew [17], a complete data set of web pages has also been used with a hierarchical classification approach to perform the classification of image-based and text-based pages. The results of these experiments are highly encouraging and show an improvement over a single classifier based approach. More recently, Ras et. al.. [25] have approached the problem of implementing a cascade classifier for incomplete data sets. They build classifiers which operate on non-singular complete data sets after partitioning of the data. The output of these classifiers is later combined in cascade to obtain final classification with better precision than a flat single stage classifier.

Aside from having an excellent performance over complete datasets, this type of hierarchical cascade classifiers has also shown an excellent performance over incomplete and heterogeneous data. Such is the case with the data set that we are using given that the semantic location information is incomplete (see section III). Hence, our approach is inspired in these cascade classifier frameworks that have been used in a variety of complex classification tasks over incomplete data (e.g. musical instrument classification).

III. Proposed Approach

In the following section we describe our approach. This encompasses data preprocessing that was required as well as the the architecture, methodology and the classifiers that were implemented.

A. Data Preprocessing

As seen in the example in table I, the database records contain information about the different events (T_k), user id, cell tower id, bluetooth id and a place label. The place label field values were manually entered by the users themselves. As a result, many of these entries are empty (i.e. incomplete) and also not standardized (e.g. ML is assumed to be the same as localhost.media.mit.edu-0). In order to manage this ambiguity, we have standardized the labels that could correspond to home and work prior to clustering and classification.

B. Architecture

Figure 1 shows the architecture of the proposed framework, composed of the following elements and processes:

- Raw Database: The complete dataset collected for all users in the whole experiment (call and text messaging logs, cell tower id, Bluetooth encounters, time and date, phone application usage and Internet navigation).
- Clustering: Cluster the antennas, as in [26], to relate the unlabeled ones to a given location.
- Localization Classifier: The localization process was made by training binary classifiers. Three well known classifiers were used (SVM, NB and ANN). This classification was used as a basis in order to perform a posterior contextual analysis (i.e. context awareness). The output of each classifier (*home, work* and *other*) was stored in the Profiling Database.
- Profiling Database: The contextual information (time, data and semantic location).
- User Classifier: In this stage the spatio-temporal contextual analysis and the inference of the user affiliation were performed. This is based in the fact that each group of people have associated characteristic behavior which can be identified via classifiers.

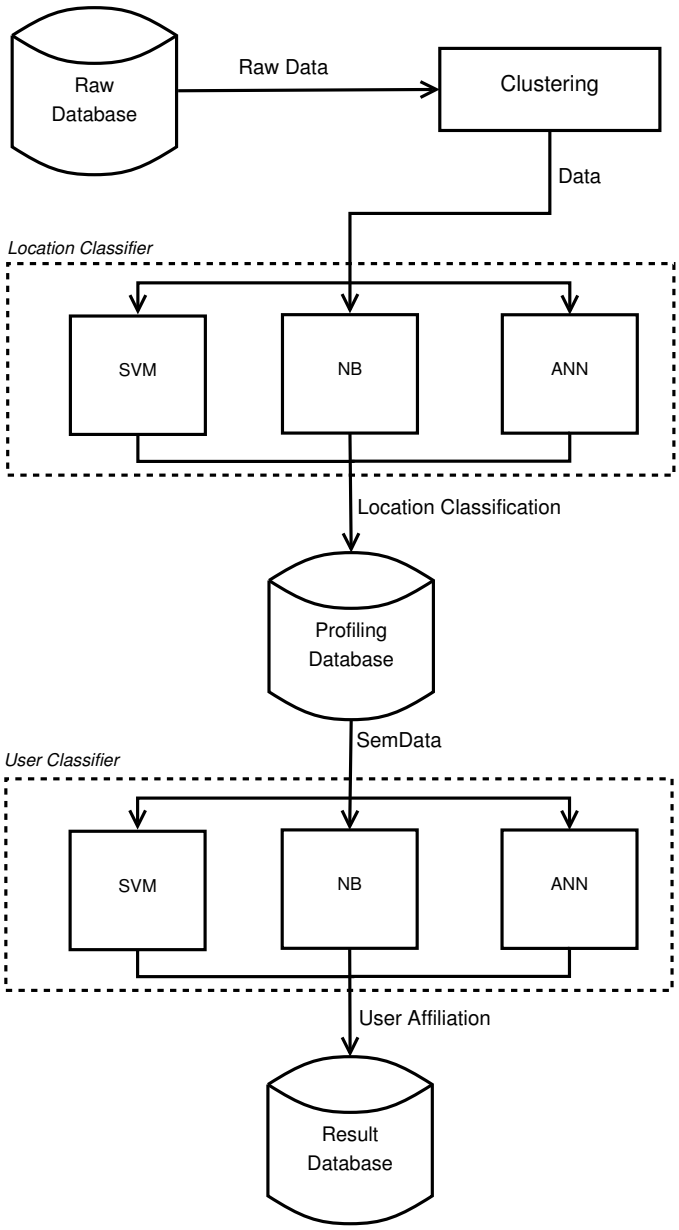


Fig. 1: Proposed framework architecture

C. Methodology

In these experiments, 94 subjects participated using an application that was preinstalled in their mobile devices over a 9 month period. The data logged, by an application in the handset, included: call logs, SMS, nearby Bluetooth devices, cellular towers ID, application usage and phone status. The users also labeled some relevant places like *home*, *work* and *else*, directly in their cellphones. The structure of the data collected is displayed in Table I, registration in a cell tower or Bluetooth encounters are the stored events.

1) Clustering: The location label was then associated to the cell tower to which the cellphone was connected at the time of labeling. There were two issues from this approach that have to be addressed, the first one was that the cell tower connection could change in time, because of signal weakening or congestion, without a change in the cellphone location, therefore one place could have a set of possible antennas associated to it. Secondly, the user could not input the labels any time the cellphone was connected

Event	User id	Tower id	Bluetooth id	Place label	
T_1	94	1		ML	
T_2	95	1		ML	
T_3	95	46		Mit	
T_4	16		1393	deskjet 5600	Work
⋮	94		120	localhost.media.mit.edu-0	
	75		438	studies.media.mit.edu	
T_k	95	289		Home	
	95	75		Home	Home
	94	46		Diesel cafe	
⋮	95	59		Mgh	
	36	5336			Else
	95		20747	BlackBerry 7100	
T_n	95	22124			

TABLE I: Raw data

to an unknown antenna, leaving a huge number of antennas without labels.

To address the first problem, we used the approach presented in [26] to cluster the cell towers to the locations *home*, *work* and *else* for each user. The second problem was addressed by training a classifier to learn from the labeled data and then apply it to the unlabeled ones. In these last tasks, we used the leverage of Bluetooth encounter data as explained in the next section. Finally, we used the affiliation information of the subjects participating in the experiments. Participants were composed of: staff members, Sloan business school alums, undergraduate students and graduate students.

2) Location Classifier: In the localization classification stage showed in figure 1, the classifier is trained with the data formatted as the following vector:

$$Data = (W_{day}, H_0, \ldots, H_{23}, T_1, \ldots, T_j, B_1, \ldots, B_k) \quad (1)$$

Where:

W_{day} = Represents if the day is a week day. This is equal to 1 if the day is a week day or 0 if otherwise.

H_i = Represents the hours of the day. Is equal to a 1 when the datum corresponds to certain hour and 0 for all other remaining hours.

T_j = Represents all the towers that the user connects to during the complete time period. Has a value equal to a 1 for all the towers to which he connects in a certain hour and is equal to a 0 for all the remaining towers.

B_k = Represents all the Bluetooth devices that the user connects to during the complete time period. Has a value equal to a 1 for all devices to which he connects to in a certain hour and is equal to a 0 for all the remaining devices. In this representation the tower id and Bluetooth identifiers represents a nominal kind of data. However, SVM can only take real valued data. Each tower id and Bluetooth idenfiers data were transformed into numerical data by associating a synthetic variable that can take values $\{0, 1\}$ depending if the corresponding tower id or Bluetooth identifier is present for a particular stored event [27]. This transformation correspond to the $T_1, \ldots, T_j$ and $B_1, \ldots, B_k$ dimensions of the vector in (1).

Once the classifier is trained, we apply the location classifier to the unlabeled data in the Raw Database to complete the missing semantic location information. Once the data set is semantically complete, it is stored in the Profiling Database as seen in figure 1.

3) User Classifier: We split the Profiling Database into training and test sets and we train the User Classifier; we then test the User Classifier using the obtained test set.

In the User Classification stage, we use the following vector:

$$SemData = (D_1, \ldots, D_7, H_0, \ldots, H_{23}, P_1, \ldots, P_3) \quad (2)$$

Where:

D_i = Represents the day of the week.

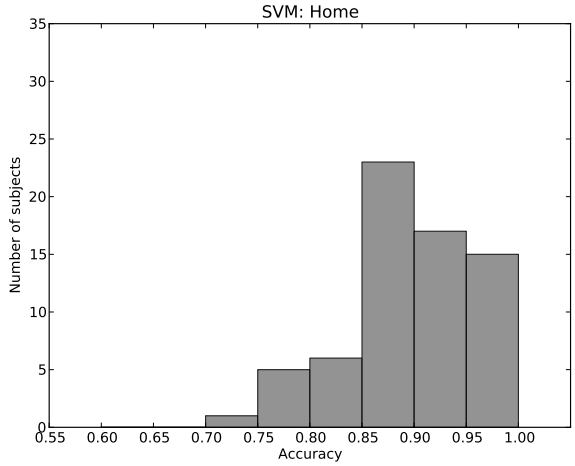
H_j = Represents the hours of the day. It is equal to a 1 when the data record corresponds to a certain hour, and 0 for all other hours.

P_k = Represents the locations obtained during the prior classification. Has a value of 1 when we have found a certain location and a 0 for remaining locations.

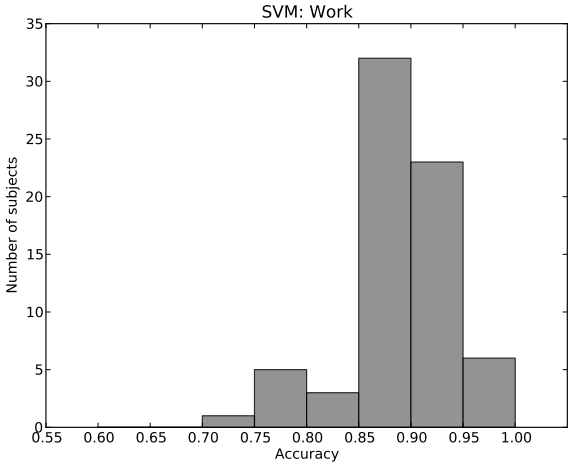
D. Implemented Classifiers

Three classifiers (i.e. SVM, Naive Bayes and Artificial Neural Networks) where trained for each individual in the dataset in the localization stage and each classifier was trained for each affiliation class in the second stage. 30% data items were used to train each first stage while 30% data items were used for testing. In the second stage, 30% semantic data items were used for training while 70% were used for testing. For the case of classification with multiple classes, we utilized the method known as OVA (One Versus All) [28], in which the class of interest has a value of 1 and the remaining classes have the value of 0.

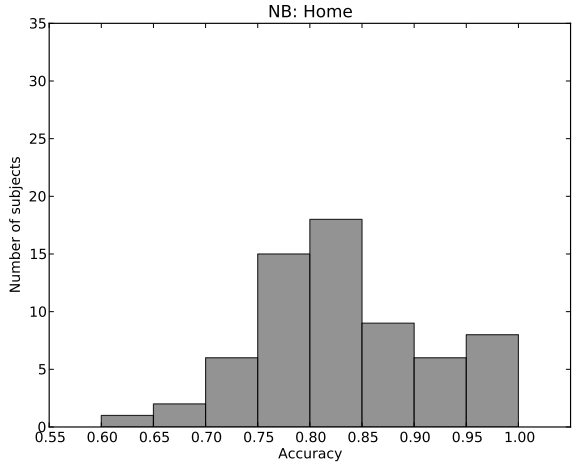
1) Support Vector Machine Classifier: The SVM classifier used the Gaussian kernel, implemented in the freely available library *libsvm*. The kernel parameter γ and the C parameter where chosen using grid-search and cross validation [29].



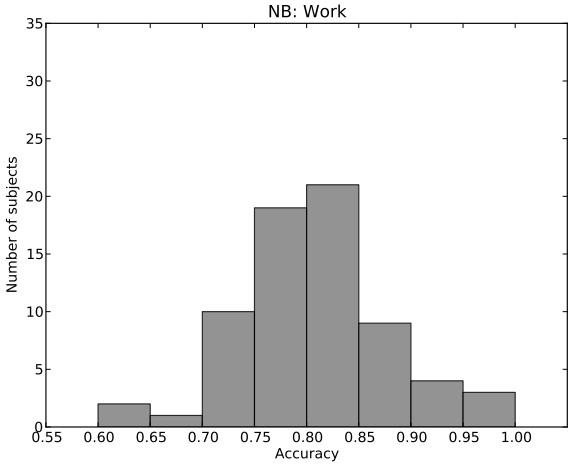
(a) SVM: *Home* Classification



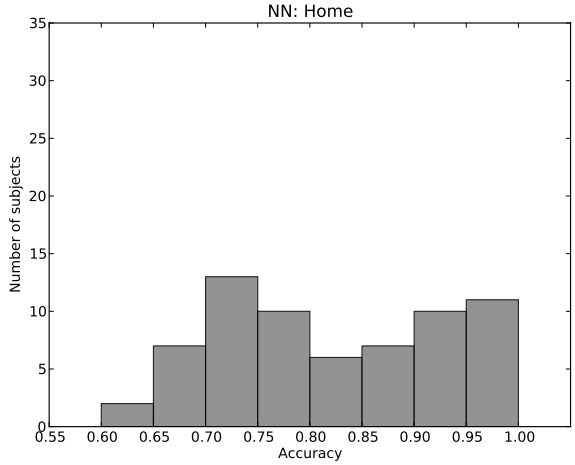
(b) SVM: *Work* Classification



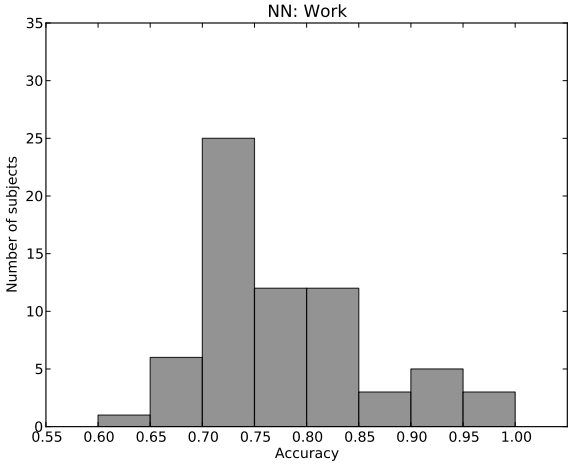
(c) NB: *Home* Classification



(d) NB: *Work* Classification



(e) NN: *Home* Classification



(f) NN: *Work* Classification

Fig. 2: Location Classification Results

2) Naive Bayes Classifier: A naive Bayes classifier is a probabilistic classifier based on Bayes Theorem under independent random feature assumptions. This classifier assumes that the presence or absence of a particular feature of a class is unrelated to any other feature (i.e. a naive assumption). Naive Bayes classifiers have worked well in many complex real-world situations [36].

3) Artificial Neural Network Classifier: Artificial neural networks (ANNs) are classifiers based upon biological neural systems [30], [31]. The model utilized is a completely connected feed-forward artificial network with one hidden layer. We have used the backpropagation learning algorithm [35] using a learning rate set at 0.001. In order to tune the network architecture, we have evaluated various hidden layer configurations and have obtained the best results with one neuron in the hidden layer in the case of the localization classifier and twenty neurons for the user classification problem.
To avoid over-training we utilized cross validation random sub-sampling as used in [33],[34]. Once training was finalized for both classifiers we utilized a different test set in order to evaluate final performance. For each case under study we performed 50 separate training runs, each of these with different random weights and selections from the validation set. Final results shown correspond to an average over the 50 training/validation runs. These results showed less than 2% error with a 95% confidence interval.

IV. Results

A. Location Classification Results

Table II and figures 2a and 2b show the results obtained classifying the *home* and *work* locations respectively using SVM. Similarly, figures 2c and 2d show the results of using NB for *home* and *work* respectively. In figure 2e, we present the results obtained by the NN used toward the classification of the data corresponding to *home*. Figure 2f shows the results of using NN for the data corresponding to *work*.

	Home	Work
NB	0.8479 ± 0.0920	0.8282 ± 0.0797
NN	0.8457 ± 0.1077	0.8022 ± 0.0801
SVM	0.9216 ± 0.0588	0.9143 ± 0.0495

TABLE II: Location classification results summary.

B. User Classification Results

In figure 3, we show the results of user classification. More detailed results can be seen in table III.

	SVM	NB	NN
Staff	92.166%	92.169%	92.145%
Sloan	75.661%	75.688%	70.547%
Undergraduate	68.021%	60.838%	70.547%
Postgraduate	79.082%	79.147%	56.045%

TABLE III: Results for user classification.

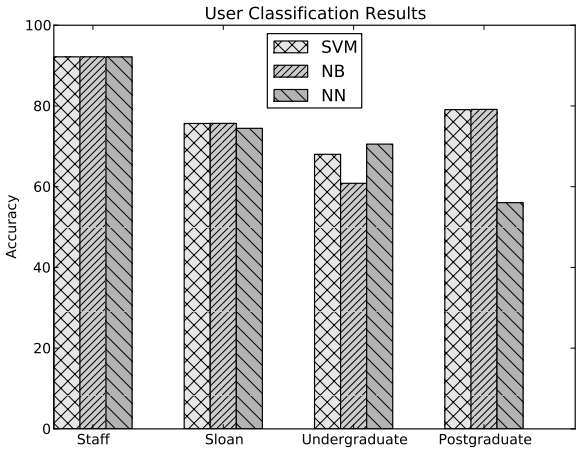


Fig. 3: User classification results for each of the tested classifiers.

V. Conclusions and Future Work

In this study, we have shown that it is possible to infer a user's affiliation using data derived from normal cellphone usage. We used clustering in combination with binary classifiers (SVM, NB, NN) to extract semantic locations (i.e. home, work and other) from cellphone and Bluetooth encounters data. These locations were used as inputs to train classifiers but this time with data corresponding to a type of user and not of an individual user as was previously done by other researchers. The location classifiers showed high accuracy across most of the users in the dataset, with SVM showing the best performance. Interestingly enough, the three classifiers had a very similar performance in the user classification stage. All of them had near 92% of successful prediction for the Staff users. This can be explained by the fact that this is notoriously the most different affiliation from the others. It could even be argued that the Staff affiliation corresponds to the only clearly different type from the other classes. A similar reasoning can be conducted to explain the lower accuracy obtained for Undergraduate, as this affiliation can be easily confused by either Sloan Business or Postgraduate.
The results showed here are biased by the nature of the dataset, and clearly does not constitute a representative sample of the variety of types and affiliations that can be found in society as a whole. However, they illustrate how, with very limited, incomplete and noisy contextual data, it is possible to discriminate from different groups of people. Future works may include the leveraging of richer contextual data to infer more precise or even more subtle affiliations of a group of people.

Acknowledgments

This research was financed by the Universidad Técnica Federico Santa María by the DGIP Grants 231021 and 231138. We want to thank Nathan Eagle from MIT for kindly

providing the Reality Mining Dataset and the incumbent documentation.

REFERENCES

[1] Measuring the Information Society: The ICT Development Index. International Telecommunication Union, 2010, ISBN 92-61-13111-5.

[2] M. Raento, A. Oulasvirta and N. Eagle, Smartphones: an Emerging Tool for Social Scientist. Sociological Methods Research, 2009, vol. 37, no. 3, pp. 426-454.

[3] Lin Liao, Donald J. Patterson, Dieter Fox, and Henry Kautz. 2007. Learning and inferring transportation routines. Artificial Intelligence 171, 5-6 (April 2007), pp. 311-331. doi=10.1016/j.artint.2007.01.006.

[4] M. González, C. Hidalgo and A. Barabási, Understanding Individual Human Mobility Patterns, Nature, vol. 453, 5 June 2008, pp. 779-782.

[5] N. Eagle, J. Quinn and A. Clauset, Methodologies for Continuous Cellular Tower Data Analysis. Pervasive Computing, LNCS, 2009, Vol 5538, pp.342-353.

[6] D. Ashbrook and T. Starner, Using GPS to learn significant locations and predict movement across multiple users, Personal and Ubiquitous Computing, Vol. 7, No. 5 , October, 2003, pp. 1617-4909.

[7] N. Eagle and A. Pentland .Reality Mining: Sensing Complex Social Systems. Personal and Ubiquitous Computing, vol 10, 4, pp. 255-268.

[8] N. Eagle, A. Pentland and D. Lazer, Inferring Social Network Structure using Mobile Phone Data, Proceedings of the National Academy of Sciences (PNAS), 2009, Vol 106 (36), pp. 15274-15278.

[9] N. Eagle and A. S. Pentland, Eigenbehaviors: identifying structure in routine, Behavioral Ecology Sociobiology, 2009, Vol 63 (7), pp. 10571066.

[10] K. Farrahi and D. Gatica-Perez, What Did You Do Today? Discovering Daily Routines from Large-Scale Mobile Data, Proceeding of the 16th ACM international conference on Multimedia, 2008, pp. 849-852.

[11] B. Boser, I. Guyon and V. Vapnik , A training algorithm for optimal margin classifiers, ACM COLT '92: Proceedings of the fifth annual workshop on Computational learning theory, pp. 144–152, 1992.

[12] C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2007.

[13] T. Mitchell, Machine Learning, McGraw-Hill, 1997.

[14] T. Huynh, M. Fritz and B. Schiele, Discovery of activity patterns using topic models, Proceedings of the IEEE International Simposium on Wereable Computers (ISWC'05), IEEE, 2005, pp. 216-222.

[15] P. Viola and M. Jones, Rapid object detection using a boosted cascade of somple features, In Computer Vision and Pattern Recognition,IEEE Computer Society, 2001, Vol I, pp. 511518.

[16] Y. Freund and R. Shapire, Experiments with a New Boosting Algorithm, International Conference on Machine Learning, 1996.

[17] C. Lu and M. Drew, Construction of a hierarchical classifier schema using a combination of text-based and image-based approaches, in SIGIR'01 Proceedings, ACM Publications, 331-336.

[18] T. Huynh, M. Fritz and B. Schiele, Discovery of activity patterns using topics models. Proceedings of the 10th International Conference on Ubiquitous Computing (UbiComp), ACM, 2008, pp. 10-19.

[19] T. Choudhury and A. Pentland. Sensing and modeling human networks using the sociometer. Proceedings of the 7th IEEE International Symposium on Wereable Computers (ISWC'05), IEEE, 2005, pp. 216-222.

[20] F. Meneses and A. Moreira. Using GSM CellId positioning for place discoveing. Proceedings of the 1st Workshop on Location Based Services for Health Care (Locare), 2006, pp. 34-42.

[21] J. Hightower, S. Consolvo, A. Lamarca, I. Smith and J. Hughes. Learning and Recognizing the Places We Go. Proceeding of the International Conference on Ubiquitous Computing (UBICOMP), Springer-Verlag, 2005, 3660, pp. 159-176.

[22] C. Zhou, D. Frankowski, P. Ludford, S. Shekhar and L. Terveen. Discovery Personally Meaningful Places: An Interactive Clustering Approach. ACM Transactions on Information Systems, 2007, vol. 25, 12.

[23] P. Nurmi and S. Bhattacharya. Identifying Meaningful Places: The Nonparametric Way. Proceedings of the 6th International Conference on Pervasive Computing, Springer, 2008, vol 5013, pp. 111-127.

[24] F. Calabrese, F. C. Pereira, G. D. Lorenzo, L. Liu and C. Latti. The Geography of Taste: Analyzing Cell-Phone Mobility and Social Events. Proceedings of the 8th International Conference on Pervasive Computing, Springer, 2010, 6030, pp. 22-37.

[25] Z. Raś, A. Dardziśka and W. Wenxin, Cascade Classifiers for Hierarchical Decision Systems, Advances in Machine Learning I , 2010, vol 262, pp. 247-256, Springer Berlin / Heidelberg.

[26] M.A. Bayir, M. Demirbas and N. Eagle (2009), Mobility Profiler: A framework for discovering mobility profiles, Pervasive and Mobile Computing, 2010, vol. 6, pp. 435-454.

[27] I. Witten and E. Frank. Data Mining: Practical Machine Learning Tools and Techniques (Second Edition). Elsevier, 2005.

[28] R. Rifkin and A. Klautau, Parallel networks that learn to pronounce english text, Journal of Machine Learning Research, pages 101141, 2004.

[29] Chih-Wei Hsu, Chih-Chung Chang and Chih-Jen Lin, A Practical Guide to Support Vector Classification, Department of Computer Science, National Taiwan University, 2010.

[30] J.-S. Roger Jang, Chuen-Tsai Sun and Eiji Mizutani, Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, Prentice Hall, 1997.

[31] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Networks, Vol. 2, No. 5., pp. 359-366., 1989.

[32] R. Hecht-Nielsen, Theory of the backpropagation neural network, International Joint Conference on In Neural Networks, 1989.

[33] R. Bouckaert and E. Frank, Evaluating the replicability of significance tests for comparing learning algorithms, Proceedings 8th Pacific-Asia Conference, 2004.

[34] Ron Kohavi, A Study of Cross-Validation and Bootstrap, In IJCAI, pp. 1137-1145. 1995.

[35] P. Werbos, Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences, Harvard University, Cambridge, MA, 1974.

[36] Harry Zhang, The Optimality of Naive Bayes, FLAIRS Conference, AAAI Press, 2004.

Cycles, Diversity and Competition in Rock-Paper-Scissors-Lizard-Spock Spatial Game Agent Simulations

K.A. Hawick
Computer Science, Institute for Information and Mathematical Sciences,
Massey University, North Shore 102-904, Auckland, New Zealand
email: k.a.hawick@massey.ac.nz
Tel: +64 9 414 0800 Fax: +64 9 441 8181

ABSTRACT
The emergence of complex spatial patterns in agent-based models is closely connected with the symmetries and relationships present between individual microscopic constituents. Games such as Rock-Paper-Scissors (RPS) have a closed cycle relationship amongst players which extends the symmetry. RPS and related games can be played by agents arranged on a spatial grid and have been shown to generate many complex spatial patterns. We consider the implications of extending the individual RPS game complexity to five-cycle games such as "Rock-Paper-Scissors-Lizard-Spock" that have competing cyclic reactions. We simulate large spatial systems using a reaction-rate formulation which are simulated for long run times to capture the dynamic equilibrium regime. We report on the stable and unstable phase mixtures that arise in these agent models and comment on the effects that drive them.

KEY WORDS
rock paper scissors lizard Spock; game theory; agents; spatial complexity; emergence.

1 Introduction

Spatial games as played by software agents have proved a useful platform for studies of complexity, with considerable activity reported very recently in the research literature [1]. The rock-paper-scissors (RPS) game [2] exhibits the closed 3-cyclic relationship of "rock blunts scissors cuts paper wraps rock" and can be played by spatial (software) agents against other agents within their neighbourhood. The rock-paper-scissors-lizard-Spock (RPSLS) variation [3] has five player states and two key competing relationship 5-cycles as explained in Section 2. The RPSLS [4] game was invented by Kass but was brought to popular attention through its appearance on an episode of the television show "The Big Bang Theory" [5]. RPSLS has considerably higher intrinsic agent-player complexity than RPS and this manifests itself in the spatial spiral patterns that emerge in simulated systems. Figure 1 shows a typical spatial pattern arising in these models. A system of spatial agents is initialised randomly and is subsequently evolved in simulation time according to microscopically simple probabilistic rules.

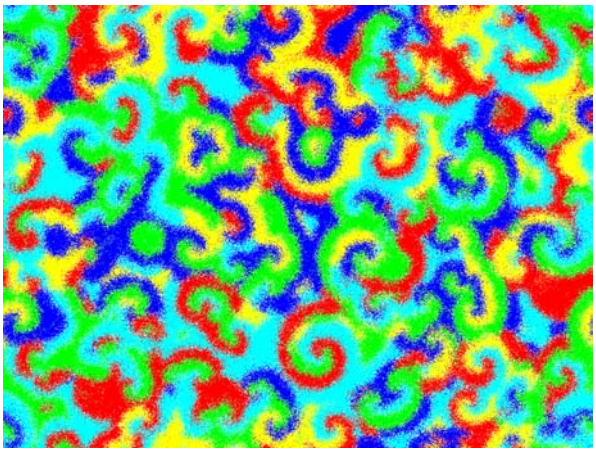


Figure 1: Snapshot configuration of the Rock, Paper, Scissors, Lizard, Spock! game on a 1024×768 spatial mesh, 2048 steps after a random start.

Gillespie formulated an approach for simulating discrete systems with stochastic rate equations [6, 7] and this method been developed further by several research groups including those of Reichenbach *et al*, Peltomaki *et al* and Szabo, Solnaki *et al* [8]. The method has been used to investigate model phase diagrams [9]; the effect of cyclic dominance [10], effects of asymmetric mobility [11] and asymmetric exclusion processes [12] such as [13].

In this present work we extend the rate equation approach used for RPS-like games to the case of competing cyclic relationships. We add an additional probabilistic rate equation to allow modelling the of RPSLS system. For this paper we report on two dimensional spatial agent game systems.

This paper is structured as follows: in Section 2 we sum-

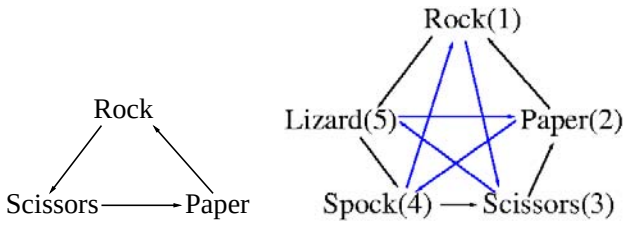


Figure 2: 3-Cycle Rock-Paper-Scissors (left) and 5-Cycle Rock-Paper-Scissors-Lizard-Spock (right).

marise key ideas for formulating a set of agents playing games like RPS or RPSLS. In Section 3 we describe how the probabilistic rate equations are established, including the competing cyclic relationships needed for agents to play RPSLS. We describe some metrics to apply to the model simulations to characterise their behaviour in Section 4 and present some systems measurements and emergent properties from the simulations in Section 5 alongwith some discussion of their implications in Section 6. We also offer some conclusions and suggested areas for further study.

2 Game Formulation

Games such as rock-paper-scissors are traditionally played by two or more players simultaneously [14] declaring their choice of the three possible entities and with a scored point based on the cyclic precedence rules below. For our purposes in this present work we focus only on two-simultaneous-player games although an agent plays the game against all its neighbouring agents in turn over time.

The traditional "rock-paper-scissors" rules are usually taken to be:

scissors cuts paper

paper wraps rock

rock bluntens scissors

with the verb in each case essentially denoting "beats," and the single resulting cycle is shown in figure 2(left).

The game was often played iteratively such as best out of three or five or some-such arrangement. As Kass has noted [3], playing RPS with opponents you knew well, often resulted in a stalemate from over-familiarity with their likely strategy. The game is therefore complicated and made considerably more "interesting" by increasing the number of strategy choices from three to five.

Kass describes the "rock-paper-scissors-Spock-lizard" rules as:

scissors cuts paper

paper covers rock

rock crushes lizard

lizard poisons Spock

Spock smashes scissors

scissors decapitates lizard

lizard eats paper

paper disproves Spock

Spock vapourises rock

rock crushes scissors

which describe two 5-cycles, which are shown in figure 2(right).

For the purposes of human players it is important the rules be easily remembered. It become difficult therefore to imagine a practical game played by humans of much more than Kass's five-entity game RPSLS. A computer however need have no such limitations and we can explore the implications of arbitrarily high cyclic games. To that end it is easier to develop a formulation based on numbered states: $1, 2, 3...$, which is a useful notation to analyse the cycles present.

The RPS Graph has a single circuit or loop of length 3. Each node in the graph has an in-degree of 1 and an out-degree of 1, and this a total degree of 2. The RPSLS graph has richer behaviour, with each node having a total degree of 4 consisting of 2 inputs and 2 outputs. There are 12 circuits, the two longest being of length 5:

```
1 3 2 1          1 5 4 1
1 3 2 4 1        1 5 2 1
1 3 5 4 1        1 5 2 4 1
1 3 5 2 1        3 2 4 3
1 3 5 2 4 1      3 5 4 3
1 5 4 3 2 1      3 5 2 4 3
```

where we have numbered the nodes: rock=1; paper=2; scissors=3; Spock=4; lizard=5.

Obviously a single node and a 2-node system cannot sustain a sensible cyclic relation at all. As we have seen a 3-node system has one and only one cycle. The number of possible cycles grows rapidly with the number of player strategies or states.

3 Spatial Game Formulation

We consider a model system with a number of Q states or player strategies, labelled $V, A, B, C, ...$ where we use V to denote a vacancy or empty site and the letters $A, B, C, ...$ to label the number $N_s \equiv Q-1$ of different species of player agent. This notation conveniently maps easily and directly onto the $1, 2, 3...Q$ notation used to describe the cycles in the previous section. Following Reichenbach *et al* [15], we first consider the probabilistic reaction equations:

$$\begin{aligned} AE &\rightarrow AV \text{ with rate } \mu \\ BA &\rightarrow BV \text{ with rate } \mu \\ CB &\rightarrow CV \text{ with rate } \mu \\ DC &\rightarrow DV \text{ with rate } \mu \\ ED &\rightarrow EV \text{ with rate } \mu \end{aligned} \quad (1)$$

and:

$$xV \rightarrow xx \text{ with rate } \sigma \quad (2)$$

where we use x to denote any $x \in \{A, B, C, ...\}$. In addition species can move around by exchanging positions with neighbours using:

$$xy \rightarrow yx \text{ with rate } \epsilon \quad (3)$$

where $x, y \in \{A, B, C, ...\}, x \neq y$. In each case the arrow denotes an update rule that can be followed when a site and a neighbouring site are selected at random.

The system thus has a total density $\rho = 1-v = a+b+c+...$ and is the fraction of sites occupied by non-vacancies. We model a lattice of interacting agents of total size $N = L \times L$, which is initially populated by a uniform and random mix of the Q different states. This model is evolved using the rate equations in time and we can measure a number of bulk quantities as described in

We have introduced a new rate equation so as to separately model the inner blue cycle as shown in Figure 2.

$$\begin{aligned} AC &\rightarrow AV \text{ with rate } \alpha \\ BD &\rightarrow BV \text{ with rate } \alpha \\ CE &\rightarrow CV \text{ with rate } \alpha \\ DA &\rightarrow DV \text{ with rate } \alpha \\ EB &\rightarrow EV \text{ with rate } \alpha \end{aligned} \quad (4)$$

The processes and rate equations are parameterised by: μ, α which control the (two) selection rates at which one species consumes another; σ which controls the reproduction process when a species expands into vacant sites; and the diffusion constant ϵ which governs how quickly species can move around the spatial model system. For the $d = 2$ dimensional systems and the work reported here we use $\sigma = \mu = \alpha = 1, \epsilon = 2d + 1 = 5$ as base vales, with some parameter scans done in α, ϵ. Once a site and neighbour are randomly chosen, the process is to be followed is determined stochastically by normalising the rates so that the probability of forward selection for example is $\frac{\mu}{\mu+\alpha+\sigma+\epsilon}$, and so forth.

In principle one could contrive experiments with different values of $\mu_{A,B}, \mu_{B,C}, ...$ and similarly for $\alpha_{A,C}, \alpha_{B,D}, ...$ We have for simplicity set all these to a single μ or α value.

Reichenbach *et al* [15] and also Peltomaki *et al* [16] have investigated variations in ϵ for the three and four state models. In a prior work [17] we investigated the simpler RPS model with up to $Q = 14$ states or fixed rates In this present work we focus on the effects of varying the rates while fixing the number of states $Q = 6$ as appropriate for the RPSLS game.

We simulate the model on a square lattice, with most results quoted on systems of $N = 1024 \times 1024$ sites, for simulation times of ≈ 16384 steps. A simulation step is defined as on average carrying out one attempt to update each and every randomly chosen site. This algorithm avoids sweeping effects or correlation pattern artifacts that would be artificially introduced if we simply swept through sites to update in index order.

The key point to note is that although we have five player strategies in a game like RPSLS, we use six states in the spatial model. We are free to set the number of vacancies to zero but it turns out the vacancies add thermal noise to the model system and considerably speed up its dynamics [17].

In Figure 3 we show some snapshots of a typical RPSLS model configuration as it evolves on a logarithmic time scale. Following a uniform random initialisation with an equal fraction of agents of all player strategies (and vacancies) present, the system goes through two distinct temporal stages. During the first transient stage agents start to organise themselves into cooperative or competitive regions according to the two cyclic RPSLS rules and larger spatial structures - patches and waves and interleaving layers start to form. The second phase shows the development of competing spiral structures that grow in spatial extent with time and is characterised by a dynamic equilibrium reached amongst the agent players as they follow the rate equations.

Shown alongside the configuration snapshots are the location of the vacancies. Tracking the 'V' sites gives insight into where the current spatial regions of activity and fluctuations are. In a previous work the vacancies were shown to be crucial in determining the symmetry of an RPS-like game. In the figure here we see vacancies are more spread out and are not just located at domain boundaries, but penetrate quite deeply into the spirals, waves and layers of interacting agents.

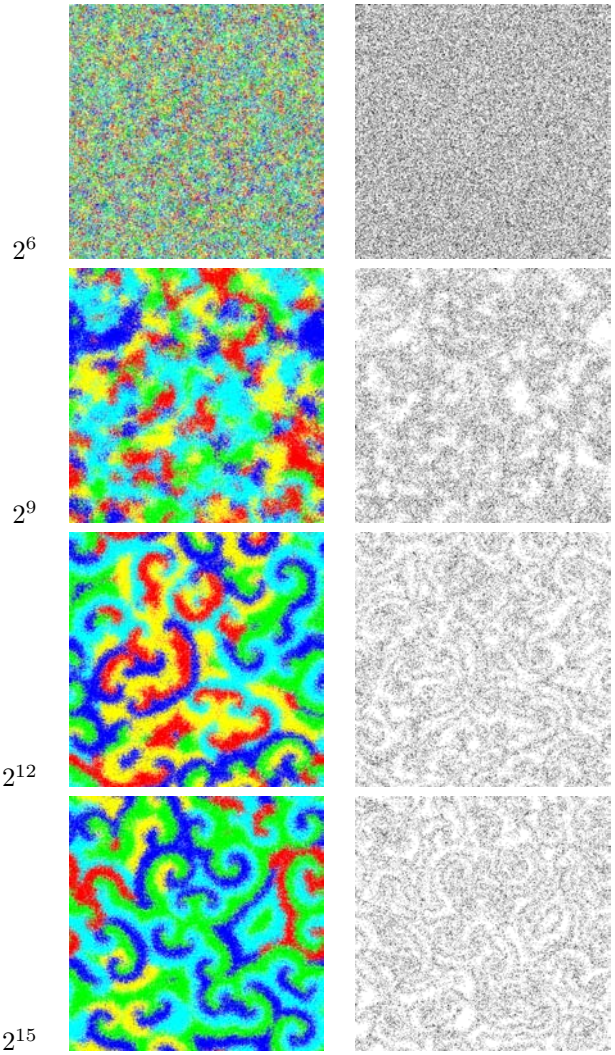


Figure 3: RPSLS agents (left) and distribution of vacancies (right) at times: 64, 512, 4096, 32768 for a 512×512 player mesh. Red=Rock; Yellow=Paper; Blue=Scissors; Green=Spock; Cyan=Lizard.

4 Metrics

There are a number of bulk properties we can measure on the system to characterise its behaviour. Simple population traces against time can be plotted, and in particular the number of species extinctions needs to be tracked. We define N_{Ex} as the number of extinctions that have occurred since the system was initialised. Similarly population fractions N_V is the number of vacancies and $N_1, N_2, ...N_Q$ can also be tracked.

A simple count of the like-like species bonds or nearest neighbour relationships is:

$$N_{\text{same}} = \frac{1}{N.d} \sum_{i=1}^{N} \sum_{j=1}^{d} 1 : s_i = s_j \qquad (5)$$

where N is the number of agent cells or sites in the d-dimensional lattice and each $s_i = 0, 1, 2, ...Q$, where we use $Q = 6$ as the number of possible states, including vacancies.

Similarly $N_{\text{diff}} = 1 - N_{\text{same}}$ is the fraction of possible bonds in the system that are different. An energy formulation would link N_{same} to an energy function through some like-like coupling term. We expect this to relax to a steady state value for a given rate equation parameter set.

The game playing rate equations are track-able via the selection or neutral fractions. We define:

$$N_{\text{sel}} = \frac{1}{N.d} \sum_{i=1}^{N} \sum_{j=1}^{d} 1 : s_i = x; s_j = x - 1, \forall x \qquad (6)$$

so it measures the fraction of agents in the system that apply the selection game playing rule. We can split this into N^{μ}_{sel} and N^{α}_{sel} for the two separate rules cycles we employ in the RPSLS game:

$$N^{\mu}_{\text{sel}} = \frac{1}{N.d} \sum_{i=1}^{N} \sum_{j=1}^{d} 1 : s_i = x; s_j = x - 1, \forall x.$$

$$N^{\alpha}_{\text{sel}} = \frac{1}{N.d} \sum_{i=1}^{N} \sum_{j=1}^{d} 1 : s_i = x; s_j = x - 2, \forall x. \qquad (7)$$

where the μ rule applies to successive player strategies around the RPSSL loop and the α rule skips a value.

We can also define N_{neut} as the remaining neutral fraction to whom the game playing rules did not apply.

5 Results

The metrics described above show the different processes that dominate during the transient and dynamic equilibrium phases observed.

Figure 4 shows how the measurement metrics typically vary with simulation time - the data is for a single run and therefore shows the unaveraged temporal fluctuations as spatial agent domains wax and wane. Providing no agent player-species die out, the metric time traces tend towards well defined dynamic equilibrium values that are controlled solely by the rate equation parameters and are

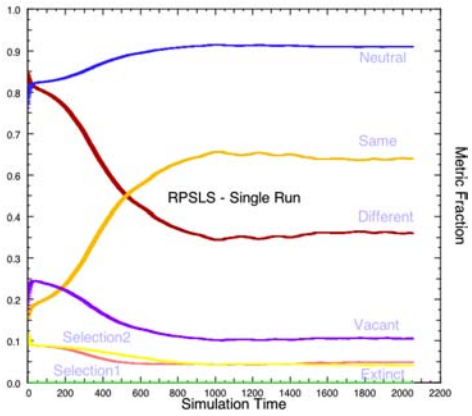


Figure 4: Metrics for a single 1024×1024 RPSLS spatial game showing the transient and dynamic equilibrium time regimes.

independent of the initial model starting conditions. We measure these long-term values – averaged over between around 100 separately initialised model runs. We can thus produce surface plots showing the various fractional metrics, with two-parameter scans on the x-y axes of the diffusion parameter ϵ and the (outer cycle) selection parameter μ Results shown are for parameter values: $\mu = \sigma = 1$ on a 1024×1024 model lattice run for 8192 equilibration steps followed by 8192 measurement steps with α, ϵ varying as shown on the axes.

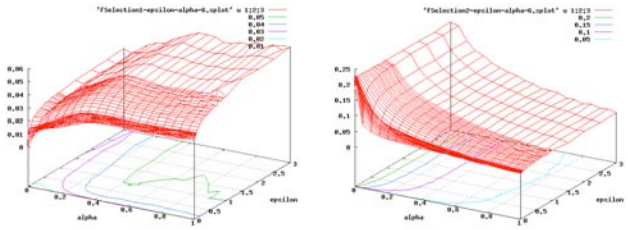


Figure 5: Fraction of Agents that have played the Selection 1 (μ, left) and 2 (α, right) RPSLS rules.

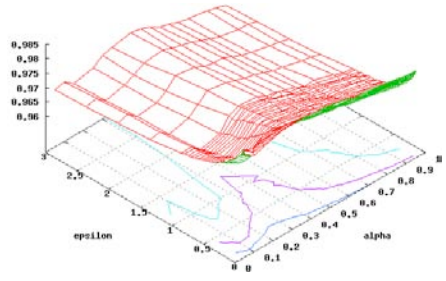


Figure 6: Fraction of Neutral Agent-Agents.

Figure 5 shows how the two selection rate equations contrast with varying diffusion mobility ϵ and by adjusting the inner cycle RPSLS interaction rate α. The two selection processes work in opposite senses but combine cause a subtle cross over effect around $\alpha^* = 0.4$ which is seen in the neutral bond count metric as shown in Figure 6.

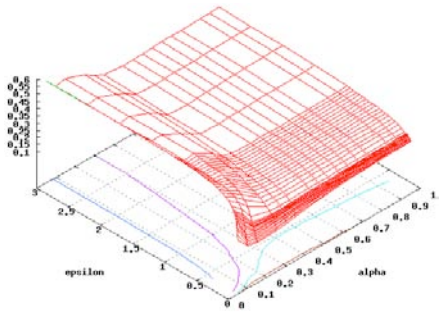


Figure 7: Fraction of Different Agent-Agent bonds.

This effect is also apparent in the surface plot of the fraction of different bonds, in Figure 7 where again a crease in the smoothly varying surface is seen at around $\alpha = 0.4$

The surface plots are all shown on the same horizontal scales although different vertical scales are used as the metric fractions do differ in their scales of interest as seem in Figure 4. Some simple contouring projections are shown underneath each surface parametric plot.

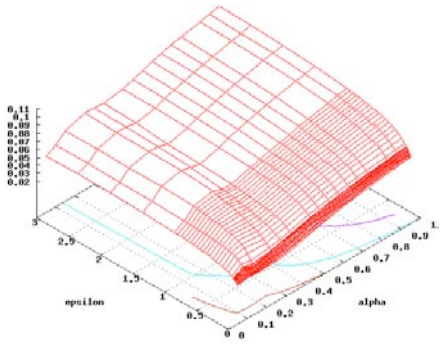


Figure 8: Fraction of Vacancies.

Figure 8 shows the long-term trend in the count of vacant sites in the RPSLS model. This is seen to drop monotonically with decreasing agent mobility, but to rise with greater activity from the inner cycle α rate equation. Once again there is a crease in the surface arising around $\alpha = 0.4$.

In an attempt to understand the symmetry relationships that arise from the competing phases we attempt to generalise the simulation by simulating systems with different numbers of initial states Q. We track some of the measurements for fixed parameter values: $\mu = \sigma = 1; \alpha = 0; \epsilon = 5$ on a 1024×1024 model lattice run for 8192 equilibration

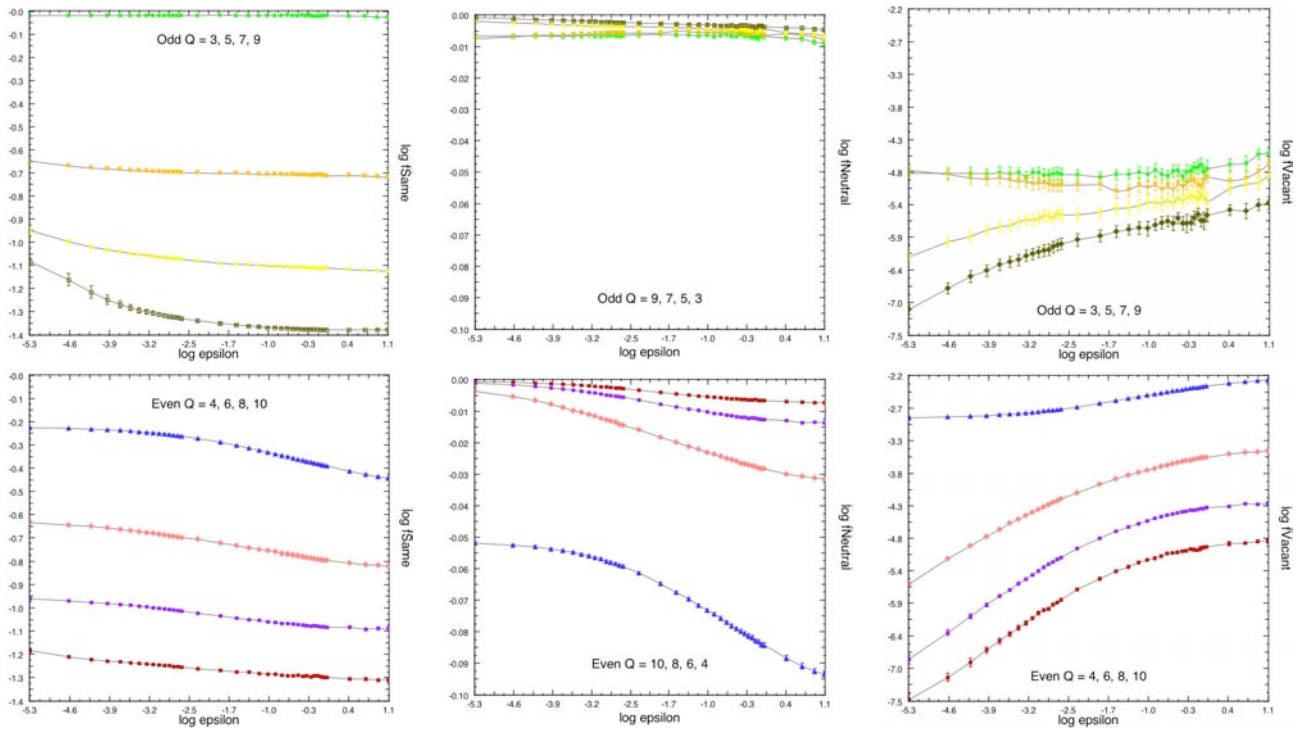


Figure 9: Fraction of Same, Neutral bonds and vacancies for odd/even Q in 1024×1024 RPSLS mesh. Q values as stated are from top to bottom.

steps followed by 8192 measurement steps. Averages were made over 100 independent runs with error bars computed from the standard deviations over these 100 samples. Setting $\alpha = 0$ simplifies the selection rate process and allows a sensible comparison to be made between different Q values.

Figure 9 shows the fractions of same-bonds; neutral bonds and of vacant sites in the RPSLS model played on 1024×1024 cells, measured over 8192 steps, with an additional initial 8192 steps discarded for equilibration from a random start. Measurements are shown for various odd and even numbers of initial states Q. There are a number of interesting effects apparent.

Most prominently there is considerable difference in qualitative behaviour for even and odd numbers of player agents. For the RPSLS case of even $Q = 6$ we see how there is a broad separation of the fraction of Like-Like agents, and Vacant site curves. The case of odd-Q shows much higher variances in the measured points. We speculate this is due to the frustrations encountered by agents arranged in an odd-Q system. For even-Q it is possible for agents to arrange themselves in alternating layers. So although the rate equations drive the system to seek like-like agents, failing that, agents can arrange to be next to a non competing species in the case of Even Q. This is not possible in the case e of odd-Q and greater spatial fluctuations arise as more player win-lose situations arise.

6 Discussion & Conclusions

We have explored the spatial agent game of "rock, paper, scissors, lizard, Spock" and have shown it exhibits considerably more complex spatial behaviour than simpler single-cycle games such as RPS and its variations. In particular it exhibits spiral patterns as seen in other unrelated spatial agent models [18].

We found that the model system exhibits a transient phase following random initialisation that is inevitably followed by a period of dynamic equilibrium. Although domain coarsening occurs during the dynamic equilibrium phase it is characterised by unchanging average values for the various metrics we have discussed. The long term values of these metrics - and their variances can then be used to characterise the system as whole for the particular rate equation parameters chosen.

A study of the agent mobility suggests further evidence for the importance of the vacancy fraction in the spatial system. We have also investigated competing effects between the two RPSLS game player cycles and have found preliminary evidence for a phase transition at a non-trivial value of rate parameter $\alpha^* \approx 0.4$

There is further evidence for the symmetry relationships and consequences of even/odd numbers of player species. We put forward the hypothesis that this is due to frustrations whereby in an even Q system agents can treat "my enemy's enemy as my friend" and arrange themselves in a less directly competitive spatial arrangement than is forced in an odd Q situation.

In the work here we were able to simulate large enough systems in two dimensions that the extinctions regime could be avoided, However, experience shows that if the system is too small, and is simulated for long enough then species extinctions do occur and apply a shock to the system which must then relax back to a dynamic equilibrium. This makes it more computationally difficult to investigate three dimensional systems, although we hoe to study these to look for dimensional dependence on the α^*.

We have employed very simple geometries for the spatial system with short range neighbourhood player interactions, but models such as the RPSLS are likely also to show interesting effects over longer range interaction geometries such as small-world links [19] and shortcuts across the spatial playing ground.

In summary the RPSLS spatial model is an exciting platform for further investigations of spatial agent complexity and emergence and holds some promise as a tool for investigating predator-prey species diversity and coexistence [20].

References

[1] Allesina, S., Levine, J.M.: A competitive network theory of species diversity. Proc. Nat. Acad. Sci. (USA) **Online Early Edition** (2011) 1–5

[2] Osborne, M.J.: An Introduction to Game Theory. Number ISBN 0-19-512895-8. Oxford Univ. Press (2004)

[3] Kass, S.: Rock Paper Scissors Spock Lizard. `http://www.samkass.com/theories/RPSSL.html` (1995) Last Visited March 2011.

[4] Wikipedia: Rock paper scissors. (`http://en.wikipedia.org/wiki/Rock-paper-scissors`) Last Visited March 2011.

[5] Cendrowski, M.: The big bang theory. TV Show (2008) Season 2 - Episode - The Lizard-Spock Expansion.

[6] Gillespie, D.T.: A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. J. Comp. Phys. **22** (1976) 403–434

[7] Gillespie, D.T.: Exact stochastic simulation of coupled chemical reactions. The Journal of Physical Chemistry **81** (1977) 2340–2361

[8] Szolnoki, A., Perc, M.c.v., Szabó, G.: Phase diagrams for three-strategy evolutionary prisoner's dilemma games on regular graphs. Phys. Rev. E **80** (2009) 056104

[9] Szabó, G., Vukov, J., Szolnoki, A.: Phase diagrams for an evolutionary prisoner's dilemma game on two-dimensional lattices. Physical Review E **72** (2005) 047107

[10] Szolnoki, A., Wang, Z., Wang, J., Zhu, X.: Dynamically generated cyclic dominance in spatial prisoner's dilemma games. Phys. Rev. E **82** (2010) 036110

[11] Venkat, S., Pleimling, M.: Mobility and asymmetry effects in one-dimensional rock-paper-scissors games. Phys. Rev. E **81** (2010) 021917

[12] Reichenbach, T., Franosch, T., Frey, E.: Exclusion processes with internal states. Phys. Rev. Lett. **97** (2006) 050603–1–4

[13] Liu, M., Hawick, K., Marsland, S., Jiang, R.: Spontaneous symmetry breaking in asymmetric exclusion process with constrained boundaries and site sharing: A monte carlo study. Physica A **389** (2010) 3870–3875

[14] Rapoport, A.: Two-Person Game Theory. Dover Publications (1966)

[15] Reichenbach, T., Mobilia, M., Frey, E.: Mobility promotes and jeopardizes biodiversity in rock-paper-scissors games. Nature **448** (2007) 1046–1049

[16] Peltomaki, M., Alava, M.: Three- and four-state rock-paper-scissors games with diffusion. Phys. Rev. E **78** (2008) 031906–1–7

[17] Hawick, K.: Complex Domain Layering in Even Odd Cyclic State Rock-Paper-Scissors Game Simulations. Technical Report CSTN-066, Massey University (2011) Submitted to IASTED Mod and Sim, MS2011, Calgary.

[18] Hawick, K.A., Scogings, C.J., James, H.A.: Defensive spiral emergence in a predator-prey model. Complexity International (2008) 1–10

[19] Zhang, G.Y., Chen, Y., Qi, W.K., Qing, S.M.: Four-state rock-paper-scissors games in constrained newman-watts networks. Phys. Rev. E **79** (2009) 062901

[20] Reichenbach, T., Mobilia, M., Frey, E.: Coexistence versus Extinction in the Stochastic cyclic Lotka-Volterra model. Phys. Rev. E **74** (2006) 051907–1–11

Artificial Bee Colony Based Power System Stabilizer Design for a Turbo-Generator in a Single-Machine Power System

H. Shayeghi
Technical Eng. Department
University of Mohaghegh Ardabili
Ardabil, Iran

H. A. Shayanfar
E.E.Department
Center of Excellence for Power System Automation and Operation
Iran University of Science and Technology,
Tehran, Iran

A. Ghasemi
Technical Eng. Department
University of Mohaghegh Ardabili
Ardabil, Iran

hshayeghi@gmail.com, hashayanfar@yahoo.com, ghasemi.agm@gmail.com

Abstract- *This paper presents an Artificial Bee Colony (ABC) algorithm for optimal tuning of the Power System Stabilizer (PSS) in a Single-Machine Infinite-Bus (SMIB) power system. The design problem of robustly tuning PSS parameters is formulated as an optimization problem according to the time domain-based objective function which is solved by the ABC technique that has strong ability to find the most optimistic results. To ensure performance and robustness of the proposed control strategy to stabilize low frequency oscillations The design process takes a wide range of operating conditions. The effectiveness of the proposed ABC based PSS is demonstrated on a SMIB power system through the nonlinear time domain simulation and some performance indices under different operating conditions in comparison with the particle swarm optimization based tuned stabilizer and conventional PSS. Results evaluation show that the proposed stabilizer achieves good robust performance for wide range of system operation conditions and is superior to the other PSSs. Moreover, the proposed control strategy has simple structure, easy to implement and tune which can be useful for the real world complex power system.*

Keywords: PSS Design, ABC Optimization, Low Frequency Oscillations, SMIB.

1. Introduction

Stability of power systems is one of the most important aspects in electric system operation .This arises from the fact that the power system must maintain frequency and voltage levels at the nominal values, under any disturbance, like a sudden increase in the load, loss of one generator or switching out of a transmission line during a fault [1]. By the development of interconnection of large electric power systems, there have been spontaneous system oscillations at very low frequencies in order of 0.2-3.0 Hz .Once started, they would continue for a long period of time. In some cases, they continue to grow, causing system separation if no adequate damping is available. Moreover, low frequency oscillations present limitations on the power-transfer capability. To enhance system damping, the generators are equipped with Power System Stabilizer (PSS) that provide supplementary feedback stabilizing signals in the excitation system. PSS augment the power system stability limit and extend the power-transfer capability by enhancing the system damping of low frequency oscillations associated with the electromechanical modes [2].

The lead compensator based stabilizers with fix parameters have practical applications and generally provide acceptable dynamic performance. However, the problem of PSS parameter tuning is a complex exercise. A number of conventional techniques have been reported in the literature pertaining to design PSS namely: the eigenvalue assignment, mathematical programming, gradient procedure for optimization and also, the modern control theory [2-5]. Unfortunately, the conventional techniques are time consuming as they are iterative and require heavy computation burden and slow convergence. In addition, the search process is susceptible to be trapped in local minima and the solution obtained may not be optimal [4]. Also, a set of controller parameters which stabilize the system under a certain operating condition may no longer yield satisfactory results when there is a drastic change in power system operating conditions and configurations [5]. A more reasonable design of the PSS is based on the gain scheduling and adaptive control theory as it takes into consideration the nonlinear and stochastic characteristics of the power systems [6-7]. This type of stabilizer can adjust its parameters on-line according to the operating condition. Many years of intensive studies have shown that the adaptive stabilizer can not only provide good damping over a wide operating range but, more importantly, it can also, solve the coordination problem among the stabilizers. Many random heuristic methods, such as Tabu search, genetic algorithms, chaotic optimization algorithm, rule based bacteria foraging and Particle Swarm Optimization (PSO) have

* Corresponding Author. E-Mail Address: hashayanfar@yahoo.com (H. A. Shayanfar)

recently received much interest for achieving high efficiency and search global optimal solution in the problem space and they have been applied to the problem of PSS design [8-11]. These evolutionary based methods are heuristic population-based search procedures that incorporate random variation and selection operators. Although, these methods seem to be good approaches for the solution of the PSS parameter optimization problem, however, when the system has a highly epistatic objective function (i.e. where parameters being optimized are highly correlated), and number of parameters to be optimized is large, then they have degraded effectiveness to obtain the global optimum solution. In order to overcome these drawbacks, an Artificial Bee Colony (ABC) algorithm is proposed for optimal tuning of PSS parameters to improve power system low frequency oscillations damping in this paper. The ABC algorithm is a typical swarm-based approach optimization, in which the search algorithm is inspired by the intelligent foraging behavior of a honey bee swarm process [12] and has emerged as a useful tool for the engineering optimization. It incorporates a flexible and well-balanced mechanism to adapt to the global and local exploration and exploitation abilities within a short computation time. Hence, this method is efficient in handling large and complex search spaces [13].

The proposed method has been applied and tested on a weakly connected power system under wide range of operating conditions to show the effectiveness and robustness of the proposed ABC based tuned PSS and their ability to provide efficient damping of the low frequency oscillations .To show the superiority of the proposed design approach, the simulations results are compared with the PSO based designed and classical PSS under different operating conditions through some performance indices. The results evaluation show that the proposed method achieves good robust performance for wide range of load changes in the presence of large disturbance and is superior to the other stabilizers.

2. Power System Model

A power system model consisting of a Single Machine connected to an Infinite Bus (SMIB) through a circuit transmission line is used in the simulation studies. A schematic diagram for the model is shown in Fig. 1. The generator is equipped with excitation system and a power system stabilizer. All the relevant parameters are given in the Appendix.

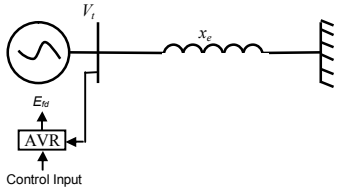


Fig. 1. SMIB power system

The synchronous generator is represented by model 1.1, i.e. with field circuit and one equivalent damper winding on the q axis. The dynamic equations of the SMIB system considered can be summarized as [14, 11]:

$$\begin{aligned} \dot{\delta} &= \omega_B S_m \\ \frac{dS_m}{dt} &= \frac{1}{2H}(-DS_m + T_m - T_e) \\ \dot{E}'_q &= \frac{1}{T'_{do}}(E_{fd} + (x_d - x'_d)i_d - E'_q) \\ \dot{E}_{fd} &= \frac{1}{T_A}(k_A(v_{ref} - v_t + V_s)) - E_{fd}) \end{aligned} \quad (1)$$

$$T_e = E'_q i_q + (x'_d - x'_q) i_d i_q \quad (2)$$

2.1. Structure of the PSS

The structure of PSS, to modulate the excitation voltage is shown in Fig. 2. The structure consists of a gain block with gain K, a signal washout block and two-stage phase compensation blocks. The input signal of the proposed method is the speed deviation ($\Delta\omega$) and the output is the stabilizing signal V_S which is added to the reference excitation system voltage. The signal washout block serves as a high-pass filter, with the time constant T_W, high enough to allow signals associated with oscillations in the input signal to pass unchanged. From the viewpoint of the washout function, the value of T_W is not critical and may be in the range of 1 to 20 seconds [11]. The phase compensation block (time constants T_1, T_2 and T_3, T_4) provides the appropriate phase-lead characteristics to compensate for the phase lag between input and the output signals.

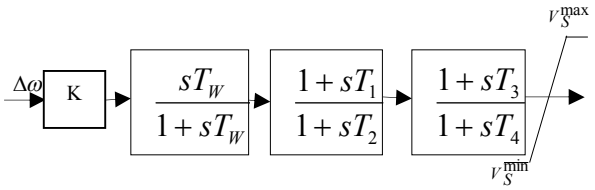


Fig .2. Structure of power system stabilizer

3. ABC Algorithm

Recently, Karaboga and Basturk [15] have described an artificial bee colony algorithm based on the foraging behavior of honey-bees for the numerical optimization problems. The algorithm simulates the intelligent foraging behavior of the honey bee swarms. It is a very simple, robust and population based stochastic optimization algorithm [16].

The minimal model of forage selection in the honey bee swarms intelligence consists of three essential components: food sources, employed foragers and unemployed foragers, and two leading modes of the behavior, recruitment to a nectar source and abandonment of a source, are defined [17]. A food source value depends on many factors, such as its

proximity to the nest, richness or concentration of energy and the ease of extracting this energy. The employed foragers are associated with particular food sources, which they are currently exploiting or are ‘‘employed”. They carry with them information about these food sources and share this information with a certain probability. There are two types of unemployed foragers, scouts and onlookers. Scouts search the environment surrounding the nest for new food sources, and onlookers wait in the nest and find a food source through the information shared by employed foragers.

In the ABC algorithm, the colony of artificial bees contains of three groups of bees: employed bees, onlookers and scouts. The food source represents a possible solution of the optimization problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. Every food source has only one employed bee. Thus, the number of employed bees or the onlooker bees is equal to the number of food sources (solutions).

An onlooker bee chooses a food source depending on the probability value associated with that food source, p_i, calculated by the following expression:

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \tag{3}$$

Where fit_i is the fitness value of the solution i evaluated by its employed bee, which is proportional to the nectar amount of the food source in the position i and SN is the number of food sources which is equal to the number of employed bees (BN). In this way, the employed bees exchange their information with the onlookers.

In order to produce a candidate food position from the old one, the ABC uses the following expression:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \tag{4}$$

Where, $k \in \{1,2,...,BN\}$ and $j \in \{1,2,...,D\}$ are randomly chosen indexes. Although k is determined randomly, it has to be different from i. ϕ_{ij} is a random number between [0, 1]. It controls the production of a neighbour food source position around x_{ij} and the modification represents the comparison of the neighbor food positions visually by the bees. Equation (4) shows that as the difference between the parameters of the x_{ij} and x_{kj} decreases, the perturbation on the position x_{ij} decreases, too. Thus, as the search approaches to the optimum solution in the search space, the step length is adaptively reduced.

The food source whose nectar is abandoned by the bees is replaced with a new food source by the scouts. In the ABC algorithm this is simulated by randomly producing a position and replacing it with the abandoned one. If a position cannot be improved further through a predetermined number of cycles called *limit* then, that food source is assumed to be abandoned.

After each candidate source position v_{ij} is produced and then, evaluated by the artificial bee, its performance is compared with that of x_{ij}. If the new food has equal or better nectar than the old source, it is replaced with the old one in the memory. Otherwise, the old one is retained. In other words, a greedy selection mechanism is employed as the selection operation between the old and the current food sources.

The main steps of the algorithm are given by [12, 17]:

i) Initialize the population of solutions and evaluate them.
ii) Produce new solutions for the employed bees, evaluate them and apply the greedy selection mechanism.
iii) Calculate the probabilities of the current sources with which they are preferred by the onlookers.
iv) Assign onlooker bees to the employed bees according to probabilities, produce new solutions and apply the greedy selection mechanism.
v) Stop the exploitation process of the sources abandoned by bees and send the scouts in the search area for discovering new food sources, randomly.
vi) Memorize the best food source found so far.
vii) If the termination condition is not satisfied, go to step 2, otherwise stop the algorithm.

It is clear from the above explanation that there are three control parameters used in the basic ABC: The number of the food sources which is equal to the number of employed or onlooker bees (SN), the value of *limit* and the Maximum Cycle Number (MCN).

4. Problem Formulation

In case of the above lead-lag structured PSS, the washout time constants is usually specified. In the present study, washout time constant T_W =10 sec is used. The controller gain K and the time constants T_1, T_2, T_3 and T_4 are to be determined. It is worth mentioning that the PSS is designed to minimize the power system oscillations after a large disturbance so as to improve the power system stability. These oscillations are reflected in the deviations in power angle, rotor speed and line power. Minimization of any one or all of the above deviations could be chosen as the objective. In this study, an Integral Square Time of Square Error (ISTSE) of the speed deviations is taken as the objective function expressed as follows:

$$J = \sum_{i=1}^{NP} \int_{t=0}^{t=tsim} t^2 (\Delta\omega)^2 \, dt \tag{5}$$

Where, $\Delta\omega$ denotes the rotor speed deviation for a set of PSS parameters, t_{sim} is the time range of the simulation and NP is the total number of operating points for which the optimization is carried out. It is

aimed to minimize this objective function in order to improve the system response in terms of the settling time and overshoots under different operating condition. The design problem can be formulated as the following constrained optimization problem, where the constraints are the controller parameters bounds [9, 11]:

Minimize J Subject to:

$$\begin{aligned} K^{\min} &\le K \le K^{\max} \\ T_1^{\min} &\le T_1 \le T_1^{\max} \\ T_2^{\min} &\le T_2 \le T_2^{\max} \\ T_3^{\min} &\le T_3 \le T_3^{\max} \\ T_4^{\min} &\le T_4 \le T_4^{\max} \end{aligned} \quad (6)$$

Typical ranges of the optimized parameters are [0.01-50] for K and [0.01-1] for T_1, T_2, T_3 and T_4. The proposed approach employs ABC algorithm to solve this optimization problem and search for an optimal or near optimal set of PSS parameters. The optimization of the PSS parameters is carried out by evaluating the objective cost function as given in Eq. (5), which considers a multiple of operating conditions are given in Table 1. The operating conditions are considered for wide range of output power at different power factors. Results of the PSS parameter set values based on the objective function J, by applying a three phase-to-ground fault for 100 ms at generator terminal at $t=1$ sec using the proposed ABC and PSO algorithms are given in Table 2. The Classical PSS (CPSS) is designed using the tuning guidelines given in [14] for the nominal operating point. Fig. 3 shows the minimum fitness functions evaluating process.

Table 1. Operation conditions

Case No.	P	Q	x_e	H
Case1 (Base Case)	0.8	0.4	0.3	3.25
Case 2	0.5	0.1	0.3	3.25
Case 3	1	0.5	0.3	3.25
Case 4	0.8	0.4	0.6	3.25
Case 5	0.5	0.1	0.6	3.25
Case 6	1	0.5	0.6	3.25
Case 7	0.8	0	0.6	3.25
Case 8	1	-0.2	0.3	3.25
Case 9	0.5	-0.2	0.6	3.25
Case 10	1	0.2	0.3	0.81

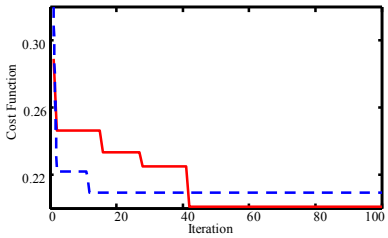


Fig. 3: Fitness convergence, Dashed (PSO) and Solid (ABC).

Table 2. Optimal PSS parameters

Method	K_{pss}	T_1	T_2	T_3	T_4
ABC	29.31	0.0941	0.0213	0.0657	0.0011
PSO	20.56	0.098	0.0195	0.0883	0.0103
CPSS	12.5	0.0738	0.0280	0.0738	0.0280

5. Simulation Results

The behavior of the proposed ABC based designed PSS (ABCPSS) under transient conditions is verified by applying disturbance and fault clearing sequence under different operating conditions. In comparison with the PSO based tuned PSS (PSOPSS) and classical PSS. The disturbances are given at t = 1 sec. System responses in the form of slip (S_m) are plotted.

Fig. 4 shows the system response at the lagging power factor operating conditions with weak transmission system by applying a step change of 0.1 pu in input mechanical torque. It can be seen that the system with CPSS is highly oscillatory. ABC and PSO based tuned stabilizers are able to damp the oscillations reasonably well and stabilize the system at all of the operating conditions. Fig. 5 depicts the responses of the same operating conditions but, with strong transmission system. System is more stable in this case, following any disturbance. Both PSSs improve its dynamic stability considerably and ABCPSS shows its superiority over PSOPSS and CPSS. Fig. 6 refers to a three-phase to ground fault for 100 ms at generator terminal. It can be seen that the proposed PSS has good performance in damping low frequency oscillations and stabilizes the system quickly. Moreover, the system performance analysis under different operating conditions and disturbances show that the ABC based tuned PSS is superior to the PSO and the classical based methods designed stabilizer.

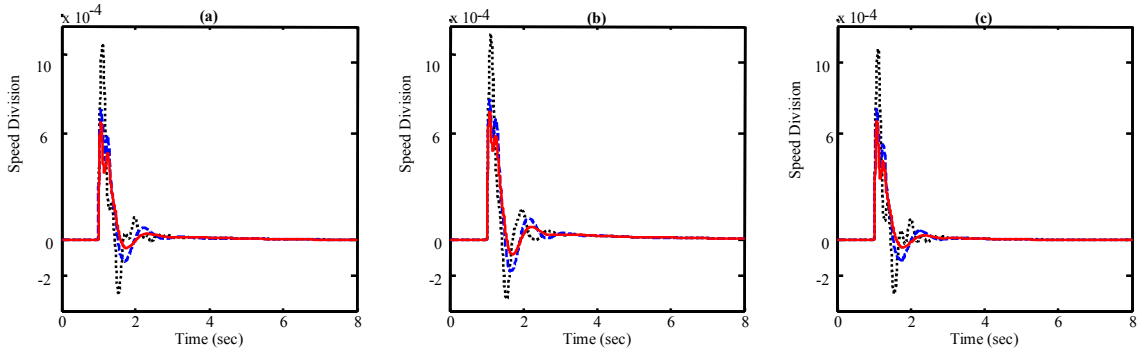


Fig. 4. ΔT_m=0.1 (p.u.), X_e=0.3; CPSS (Dotted), PSOPSS (Dashed) and ABCPSS (Solid)
a) P=0.8, Q=0.4 b) P=0.5, Q=0.1 c) P=1.0, Q=0.5

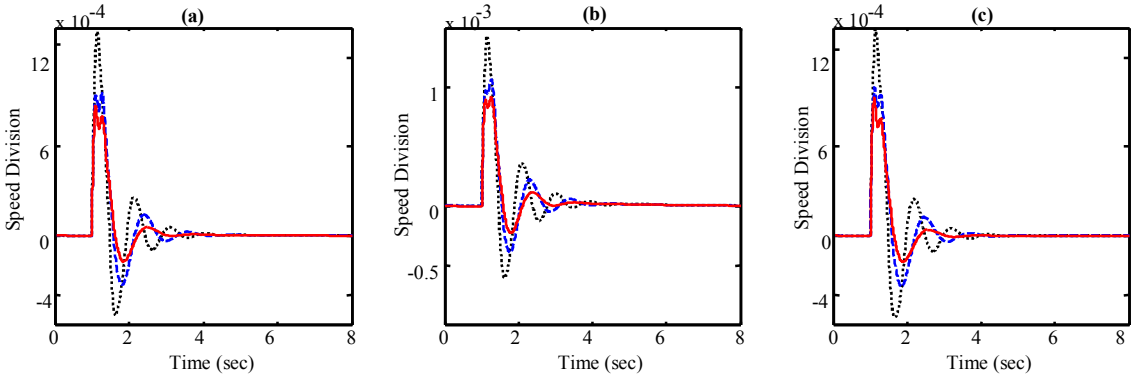


Fig. 5. ΔT_m=0.1 (p.u.), X_e=0.6; CPSS (Dotted), PSOPSS (Dashed) and ABCPSS (Solid)
a) P=0.8, Q=0.4 b) P=0.5, Q=0.1 c) P=1.0, Q=0.5

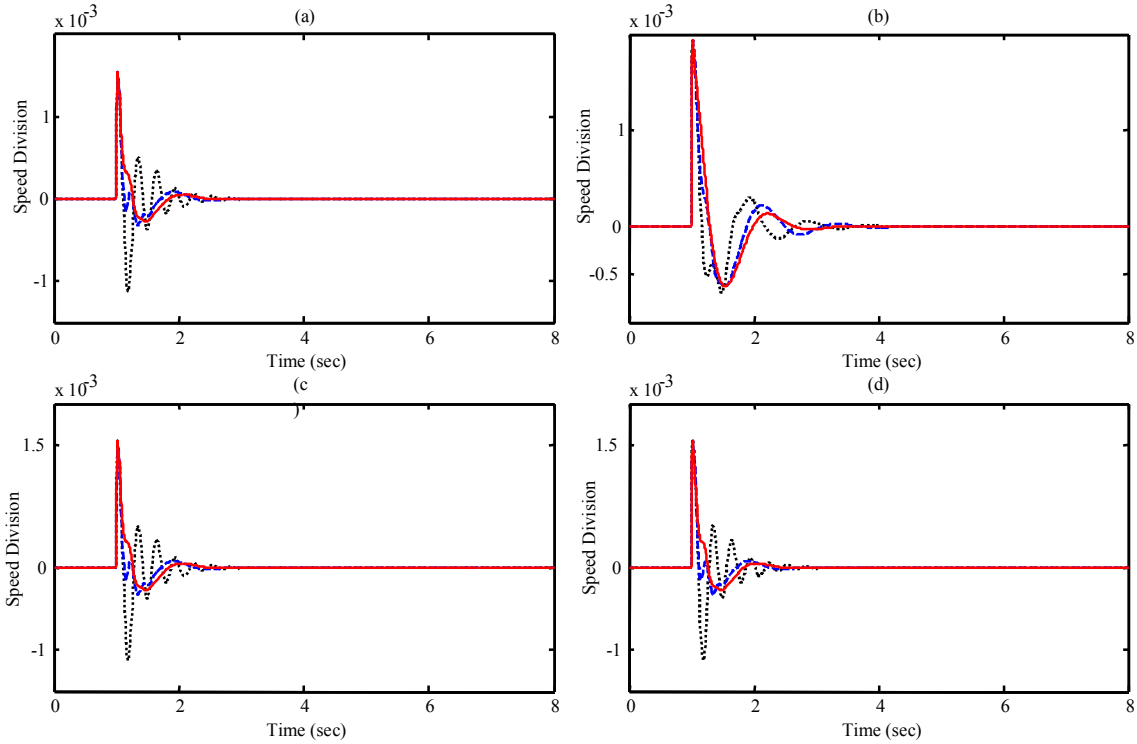


Fig. 6. 3-φ to ground fault 100 ms, CPSS (Dotted), PSOPSS (Dashed) and ABCPSS (Solid)
X_e=0.3: a) P=0.8, Q=0.4 b) P=1.0, Q=0.5 X_e=0.6: c) P=0.8, Q=0.0 d) P=0.8, Q=0.2

To demonstrate performance robustness of the proposed method, two performance indices: the Integral of the Time multiplied Absolute value of the Error (ITAE) and Figure of Demerit (FD) based on the system performance characteristics are defined as [18]:

$$FD = 10 \times [(500 \times OS)^2 + (8000 \times US)^2 + 0.01 \times T_s^2] \quad (7)$$

$$ITAE = 1000 \times \int_0^8 t \,|\Delta\omega|dt \quad (8)$$

Where, Overshoot (OS), Undershoot (US) and settling time of rotor angle deviation of machine is considered for evaluation of the FD. It is worth mentioning that the lower the value of the these indices are, the better the system response in terms of the time-domain characteristics. Numerical results of performance robustness for all cases as given in Table 1 by applying a step change of 0.1 pu in input mechanical torque at t = 1 sec are listed in Table 3.

Table 3. Performance indices

Case No	ABCPSS		PSOPSS		CPSS	
	ITAE	FD	ITAE	FD	ITAE	FD
1	0.4668	0.3823	0.5475	0.5330	0.5632	1.4729
2	0.7996	0.4561	0.8842	0.7289	0.9042	1.7696
3	0.3420	0.3830	0.4218	0.5246	0.4465	1.4774
4	0.6726	0.8191	0.9330	1.5374	1.1311	3.4747
5	1.0268	1.0293	1.2791	2.0328	1.4959	4.2209
6	0.6566	0.8854	0.9469	1.6527	1.1631	3.6312
7	0.6726	0.8191	0.9331	1.5374	1.1310	3.4747
8	0.3419	0.3828	0.4218	0.5246	0.4465	1.4774
9	1.0268	1.0293	1.2791	2.0328	1.4959	4.2209
10	0.3420	0.3830	0.4218	0.5246	0.4465	1.4774

It can be seen that the values of these system performance characteristics with the proposed ABC based tuned PSSs are much smaller compared to that of PSO and classical based designed PSS. This demonstrates that the overshoot, undershoot, settling time and speed deviations of machine is greatly reduced by applying the proposed ABC based tuned PSS.

6. Conclusions

In this paper, ABC optimization technique is proposed for power system stabilizer design in a SMIB power system. To design PSS problem, a nonlinear simulation-based objective function is developed to increase the system damping and then the ABC technique is implemented to search for the optimal stabilizer parameters. The proposed ABC algorithm is easy to implement without additional computational complexity. Thereby, experiments of this algorithm gives quite promising results. The ability to jump out the local optima, the convergence precision and speed are remarkably enhanced and thus the high precision and efficiency are achieved. The effectiveness of the proposed stabilizer, for power system stability improvement, is demonstrated by a weakly connected example power system subjected to severe disturbance. The dynamic performance of the ABC based tuned PSS has also been compared with the PSO and classical methods based designed PSS to show its superiority. The nonlinear simulation results under wide range of operating conditions show the effectiveness and robustness of the ABC based PSS ability to provide efficient damping of low frequency oscillations and its superiority to the other methods. The system performance characteristics in terms of ITAE and FD indices reveal that the proposed stabilizers demonstrates that the overshoot, undershoot, settling time and speed deviations of the machine are greatly reduced under severe disturbance conditions.

Appendix: System Data

Generator: R_a=0**,** x_d= 2.0**,** x_q = 1.91**,** x'_d=0.244**,** x'_q=0.244 **,** f = 50 Hz, T'_{do}=4.18, T'_{qo}=0.75, H=3.25,
Transmission line: R= 0, x_e = 0.3.
Exciter: K_A=50**,** T_A=0.05, E_{fdmax}=7.0, E_{fdmin}=-7.0.

References

[1] M. Anderson and A. A. Fouad, Power System Control and Stability, Ames, IA: Iowa State Univ. Press, 1977.

[2] P. Kundur, Power System Stability and Ccontrol, McGraw-Hill Inc., New York, 1994.

[3] Y. Hsu, C. Y. Hsu, design of proportional integral power system stabilizer, IEEE Trans. on power Systems, Vol. 1, No. 2, pp. 46-53, 1986.

[4] P. Kundur, M. Klein, G. J. Rogers, M. S. Zywno, Application of power system stabilizers for enhancement of overall system stability, IEEE Trans. on Power Systems, 614-626, 1989.

[5] M. J. Gibbard, Robust design of fixed-parameter power system stabilizers over a wide range of operating conditions, IEEE Trans. on Power Systems, Vol. 6, pp. 794-800, 1991.

[6] D. A. Pierre, A perspective on adaptive control of power systems, IEEE Trans. on Power Systems, Vol. 2, pp. 387-396, 1987.

[7] Y. Zhang, G. P. Chen, O. P. Malik and G. S. Hope, An artificial neural network based adaptive power system stabilizer, IEEE Trans. on Energy Conversion, Vol. 8, No. 1, pp. 71-77, 1993.

[8] Y. L. Abdel-Magid, M. A. Abido, Optimal multiobjective design of robust power system stabilizers using genetic algorithms, IEEE Trans. on Power Systems, Vol. 18, No. 3, pp. 1125 -1132, 2003.

[9] H. Shayeghi, H.A. Shayanfar, S. Jalilzadeh, A. Safari, Multi-machine power system stabilizers design using chaotic optimization algorithm, Energy Conversion and Management, Vol. 51, pp. 1572-1580, 2010.

[10] S. Mishra, M. Tripathy, J. Nanda, Multi-machine power system stabilizer design by rule based bacteria foraging, Electric Power Systems Research, Vol. 77 pp. 1595-1607, 2007.

[11] H. Shayeghi, H.A. Shayanfar, A. Safari, R. Aghmasheh, A robust PSSs design using PSO in a multi-machine environment, Energy Conversion and Management, Vol. 51, pp. 696-702, 2010.

[12] D. Karaboga, B. Akay, A comparative study of artificial bee colony algorithm, Applied Mathematics and Computation, Vol. 214, pp. 108-132, 2009.

[13] C. Zhang, D. Ouyang, J. Ning, An artificial bee colony approach for clustering, Expert Systems with Applications, Vol. 37, pp. 4761-4767, 2010.

[14] K. R. Padiyar, Power System Dynamics- Stability and Control, Second edition, BS Publications, Hyderabad, India, 2008.

[15] D. Karaboga, B. Basturk A powerful and efficient algorithm for numerical function optimization: Artificial Bee Colony (ABC) algorithm, Journal of Global Optimization, Vol. 37, pp. 459-471, 2007.

[16] S. N. Omkar, J. Senthilnath, R. Khandelwal, G. N. Naik, S. Gopalakrishnan, Artificial Bee Colony (ABC) for multi-objective design optimization of composite structures, Applied Soft Computing, Article in press (2010).

[17] C. Zhang, D. Ouyang, J. Ning, An artificial bee colony approach for clustering, Expert Systems with Applications, Vol. 37, pp. 4761-4767, 2010.

[18] H. Shayeghi, A. Jalili, H. A. Shayanfar, Multi-stage fuzzy load frequency control using PSO, Energy Conversion and Management, Vol. 49, pp. 2570-2580, 2008.

Biographies

Hossein Shayeghi received the B.S. and M.S.E. degrees in Electrical and Control Engineering in 1996 and 1998, respectively. He received his Ph. D. degree in Electrical Engineering from Iran University of Science and Technology, Tehran, Iran in 2006. Currently, he is an Associate Professor in Technical Engineering Department of University of Mohaghegh Ardabili, Ardabil, Iran. His research interests are in the Application of Robust Control, Artificial Intelligence and Heuristic Optimization Methods to Power System Control Design, Operation and Planning and Power System Restructuring. He has authored five books in Electrical Engineering area, one in Power Systems Analysis, one in MATALAB, one in Electric Circuits, one in Electric DC Machines, one in Electric Installations (all in Persian language). Also, he is co-authored of a book chapter- "A Review on Load Frequency Control Strategies" in the book "Complex Behaviour of the Distributed Generation System**:** Intelligent Management of the Renewable Energy Resources for assuring the DG System Power Quality and a Sustainable Development". He has published more than 160 papers in International Journals and Conferences Proceedings. He is a member of Iranian Association of Electrical and Electronic Engineers (IAEEE) and IEEE.and IEEE.

Heidarali Shayanfar Received the B.S. and M.S.E. Degrees in Electrical Engineering in 1973 and 1979, respectively. He received his Ph. D. Degree in Electrical Engineering from Michigan State University, U.S.A., in 1981. Currently, He is a Full Professor in Electrical Engineering Department of Iran University of Science and Technology, Tehran, Iran. His Research Interests are in the Area of Application of Artificial Intelligence to Power System Control Design, Dynamic Load Modeling, Power System Observability Studies, Voltage Collapse, Congestion Management in a Restructured Power System, Reliability Improvement in Distribution Systems and Reactive Pricing in Deregulated Power Systems. He is a Member of the Iranian Association of Electrical and Electronic Engineers and IEEE. He has published more than 350 techincal papers in the International Journals and Conferences Proceedings.

Ali Ghasemi Received the B.S. Degree in Electrical Engineering from Esfahan University of Technology, Esfan, Iran in 2009. Currently, He is a M.S.E. student in Technical Eng. Department of the University of Mohaghegh Ardabili, Ardabil, Iran. His Areas of Interest in Research are the Application of Heuristic Optimization to Power System Control.

Yet another breakout inspired infeasible subset detection in constraint satisfaction problem

Jun HU[1], **Philippe GALINIER**[2], **and Alexandre CAMINADA**[1]
[1]Département Informatique, Université de Technologie de Belfort-Montbéliard, Belfort, France
[2]Department of Computer Engineering, Polytechnique Montréal, Montréal, Canada

Abstract—

In [1], Eisenberg and Faltings proposed a breakout *[2] based hybrid approach to detect infeasible subset (IS) in constraint satisfaction problems. In this paper, we adopted this algorithm as a preprocessing procedure to reduce the over-constrained constraint satisfaction problem. Then a three procedures hybrid approach is employed to detect the irreducible infeasible subset (IIS) of identified IS. The approach consists of three procedures - critical constraint locating, critical constraint extending and critical constraint subsets satisfiability testing. The extent of the subproblem is based on the first located critical constraint and is ended while it cannot be satisfied by a Tabu Search incomplete search algorithm. Finally, the MAC (Maintaining Arc Consistency during search) method provides the infeasibility of the extracted subproblem. By adopting a hybrid approach, the tractability of satisfiability testing of real world applications becomes realistic. The experimental results demonstrate the effectiveness of the method.*

Keywords: constraint satisfaction problem, infeasible subset, infeasibility,breakout

1. Introduction

Constraint satisfaction problem (CSP) is a natural modeling technique to represent many real world problems, such as crew scheduling problem or frequency assignment problem [3]. Due to the complexity of most problems, the human beings cannot verify whether one instance is feasible or not. Moreover the sizes of real world problems are large, it's more effective if we can only verify infeasibility of their subproblems whose sizes are relatively small. While these subproblems are identified, the direct failure reasons that lead the infeasibility of the global problems are also identified. By modifying them, the original problems may become feasible.

Following the terminology introduced by Chinneck [4], the infeasible subproblems mentioned before are defined as Infeasible Subsets (IS). In recent years, The infeasible subset detection in CSP has received much attention. For example in the Frequency Assignment Problem [5], which can be conveniently modeled as constraint satisfaction problems. The telecommunication operators conceive the network by assigning the avaliable freqencies on the antennae in the telecommunication network. With the adoption of complicated interference constraints among antennae in the form of CSP, the operators attempt to manage the margin between the limited radio spectrum and the satisfaction of service quality. Occasionally, it's end up with an over-constrained network model due to the limited resources of radio spectrum. If the operators can identify a sub-network which causes the most interferences, then it is possible for them to modify the network design, like re-deploying the antennae to different locations or their directions, to change the interferences constraints of the network.

It is crucial if the identified sub-network involves minimal number of antennae, it avoids the operators modifying the non-related antennae. Such minimal size sub-network forms an Irreducible Infeasible Subset (IIS) of the network. Since the size of the IIS is usually relatively smaller than the given problem, by locating it, the tractability of satisfiability testing of the real world applications becomes realistic.

IIS detection is not very new research area, it has been heavily studied in SAT (SATisfiability problem) during the recent decade. The SAT is a special case of CSP. The term of MUC (Minimal Unsatisfiable Core) is frequently adopted. The MUC is the analogous of IIS in case of SAT. From the literature, the MUC detection primarily consists of two procedures:

- the satisfiability testing
- the critical constraints identification

The satisfiability testing verifies the satisfiability of all processed clauses. In case these examinated clauses constitute an unsatisfiable core, by removing the clauses or learning from resolution graph [6], the critical constraints potentially inside an MUS are idetnified.

In [7] and [8], the authors suggested an adaptive core search approach to extract MUC in unsatisfiable SAT. Inside the chronological backtracking, a fail-first [9] branch strategy guides the search toward hardest clauses. The search evaluates each visited clauses by its hardness of satisfaction, and labels them as candidates in one IIS or not. [6] proposed a graph learning [10] embedded DLL [11], which processes the satisfiability proofs during search and derives the MUC.

By converting the unsatisfiable formula into its complement feasible form, [12] identified the MUC by enumerating the satisfiable solutions for the complement form. [13] proposed AMUSE, an bottom up algorithm extended the unsatisfiable core from the empty clauses till the unsatisfiability was proven.

Liffiton and Sakallah [14] proposed a two-phase algorithm *CAMUS*, admitted to identify all MUC in the given unsatisfiable SAT by finding hitting set on a set of maximal satisfiable subsets (Max-SAT).

Grégoire et al. [15] proposed an approximated approach. They concentrate on recording all the empty clauses during search by finding maximal consistent sub-formula. Based on these identified empty clauses, an unsatisfiable core is constructed.

Galinier and Hertz [16] proposed Insertion, Removal and Hittingset method to detect the IIS. The Insertion method will identify one empty clause inside the violated constraint subset found by a Max-SAT method per iteration, and erase all the other empty clauses. The method is based on an assumption that the Max-SAT method would always violated exactly one clause per IIS. Finally, [17] gave a survey of MUC detection methods in SAT.

Comparing the research in SAT, there is a little amount of papers concerning the IIS detection in over-constrained CSP. [18] proposed an hybrid algorithm wcore, which consists of a weight heuristic based critical constraints selection and *deadend* detection embedded MAC (Maintaining Arc-Consistency during search). When the *deadend* occurs, the weight heuristic proposes critical constraints to be evaluated by MAC. [19] improved wcore by repelling the occurrence of backtracks during MAC, which yields better solutions and computational time.

In [20], the authors proposed a hybrid approach to identify the infeasible subset which consists of two phases. An approximate algorithm is positioned on the first phase, which iteratively proposes one by one critical constraint potentially inside an infeasible subset. On the second phase, a complete search gives the unsatisfiability proof the subset extracted subset. Benefiting on the hybrid approach, the computational time is largely reduced.

Eisenberg and Faltings [1] adopted an hybrid approach which embeds the breakout algorithm [2] inside an chronological backtracking. The Breakout algorithm identifies the critical variables (constraints, resp.) by incrementing the weights on violated variables (constraints, resp.) after each iteration of a local search. Based on the dynamic variables ordering proposed by the breakout algorithm, the backtracking gives a unsatisfiability proof on the extracted hard subproblem. Thus an infeasible subset is identified.

Recently, many evidences show the relation between IIS detection and hitting set problem [7], [14], [16]. In this paper, this relation will be clarified and a hybrid algorithm based on this theoretical experiences will be presented. With the aid of infeasible subset located by the breakout algorithm, the irreducible infeasible subset (IIS) identification procedure is greatly accelerated.

This paper is organized as follows. The section 2 will provide the necessary theoretical background, which is followed by the introduction of the proposed hybrid approach. The experimental results will be demonstrated in results section.

2. Background

Constraint satisfaction problem is a 3-tuple $P =< X, C, D >$, where X is a set of variables, D is a corresponding set of domains, C is a set of constraints applied on X. A subproblem P_{sub} is a tuple $P_{sub} =< X_{sub}, C_{sub}, D >$, where $X_{sub} \subseteq X$ and $C_{sub} \subseteq C$.

A solution s_i in solutions space S of the problem P is an assignment of all variables by assigning a value from their valid domains. A consistent solution is such an assignment which satisfies all constraints. An inconsistent solution is such an assignment which violates at least constraint. If there is no consistent solution found a problem P, then P is infeasible. Otherwise, P is feasible. Let v_i be the set of violated constraints corresponding the inconsistent solution s_i. In case of P is infeasible, it is interesting to locate an IS or IIS of the given problem.

Definition 1: An Infeasible Subset (IS) is a subproblem P_{sub} which is infeasible.

Definition 2: An Irreducible Infeasible Subset is an IS, such that if any of its constraints (resp. variables) is removed, it becomes feasible. In other words, if there exists an IS in which we couldn't find a smaller IS, then such IS is an IIS.

Let the constraints belonging to the IIS be the critical constraints, each IIS consists of a subset of the critical constraints h_j. The collection of n IIS is $H = \{h_1, h_2, \ldots, h_n\}$. An IIS is a tuple $< X(h_j), h_j, D >$, where $X(h_j)$ is the variables subset under the constraints subset h_j or the variables belonging to the same IIS respectively.

As mentioned in [16], the IIS can be defined from two different aspects - variable-oriented and constraint-oriented IIS. In FAP, we are more interested in identifying the interference zone. Thus the variable-oriented IIS will be the core of our process. While the constraint-oriented IIS is adopted to describe the proposed hybrid approach, because of the ease of explanation. The projection between constraint-oriented IIS and variable-oriented IIS will be presented to ease the gap.

A CSP can be represented as a constraint network, in which, the nodes are the variables and the arcs linking the nodes are the constraints. As the constraint network is in the form of graph, it shares the same properties in graph theory. The IIS of CSP inherits the connectivity property from graph theory:

Property 3: An IIS is a connected component in the context of constraint network. Any couple of variables or constraints of the IIS are connected by a path in the constraint network.

The IIS detection is firmly related with hitting set problem [17]. A hitting set problem (also known as transversal problem, set covering problem), is one of the key problems in the combinatory of the finite sets and the theory of diagnosis. The problem is described as follows.

Definition 4: Let a ground set M and a collection set S of subsets $S = \{s_1, s_2, \ldots, s_m\}$, where $s_j \subseteq M$, $\forall j \in m$. A hitting set (or transversal) of S is a subset $H \subseteq M$ that satisfies every set in the collection S; $\forall j = 1, \ldots, m$, we have $s_j \cap H \neq \emptyset$. A hitting set H is minimal if no proper subset of H is a hitting set.

Regarding above the definition of hitting set problem, the violated constraints subsets v_i can be modeled as the nonempty subsets s_i thus the IIS is the hitting set of the collection of subsets.

Theorem 5: Between any of violated constraints subset v_i and any of IIS constraints subset h_j, there exists the relation $v_i \cap h_j \neq \emptyset, \forall i \in m, \forall j \in n$.

Proof: In total m solutions found, for each solution s_i we have a correspondent subset of violated constraints v_i which contains all the violated constraints. We have n IIS which are inside the given over-constrained problem. Suppose there exists at least one relation holding $v_i \cap h_i = \emptyset$. By problem definition, $h_i \subseteq C$ and $v_i \subseteq C$. We have h_i is a subset of constraints set C excepting the violated constraints subset v_i, noted $h_i \subseteq C|v_i$. Based on its definition, each v_i is the subset of all violated constraints in the solution s_i, which means the solution s_i is partially consistent on the subset constraints of $C|v_i$. Thus it breaks the IIS Definition 2 – there is no consistent solution on the subset constraints h_i belonging to the IIS. ■

The hitting-set problem is itself a NP-complete problem [21]. There is no interests to connect the IIS detection with hitting set problem if the complexity cannot be reduced. Thus a special case of hitting set problem is introduced to ease this gap.

In one special case, the minimal hitting set is the union of all subsets of the collection of sets S if all the subsets are differentiated. As mentioned above, the subsets in S correspond to the subsets of violated constraints. If there is no intersection between any two subsets of violated constraints in S, then the hitting set - one IIS - can be easily obtained. In order to obtain the minimal IIS inside an over-constrained CSP, the number of subsets of all differentiated violated constraints equals the number of constraints belonging to this minimal IIS.

Considering the definition of IIS, by removing any variable or constraints, the IIS becomes feasible. If exactly one constraint is removed from each IIS of the problem, the derived problem becomes feasible [22], [23]. If the size of violated constraints subset v_i is minimal, the relation $v_i \subseteq H$ is strictly held. While in practice, without applying a complete search, there are always the constraints not belonging to the IIS inside v_i. In order to reduce this fraction, the heuristics with high effectiveness of *Minimizing-Conflicts* are recommended. In next section, an hybrid IIS detection algorithm will be detailed.

3. The proposed hybrid approach

In this section, the hybrid approach to identify a variable-oriented IIS is presented. In order to ease the description, the algorithm will be described in constraint-oriented manner, and then it is followed by a projection from constraints to variables which identifies the variable-oriented IIS based on constraint-oriented algorithm. The approach consists of two main algorithms - preprocessing and Extract. The Extract can be itself considered as an independant routine to identify the IIS in CSP. While during the expriments, it shows worse profermance without the preprocessing algorithm. Thus the preprocessing algorithm is introduced to reduce the size of problem entering to the Extract.

In order to describe the preprocessing algorithm, let's first recall the breakout algorithm. The algorithm can be described as following (see Algorithm 1):

Algorithm 1: breakout

input : a set of constraints C
output: a set of weights for each constraint $weights_C$ and solution S

$S \leftarrow \emptyset, weights_C \leftarrow \emptyset$;
$breakLevel \leftarrow$ CONSTANT;
repeat
  $[S, localMinima] \leftarrow localSearch(C, weight_C)$;
  if $localMinima$ *is true* **then**
    $C_v \leftarrow violatedCtrs(S, C)$;
    $weights_C \leftarrow addWeights(C_v, weights_C)$;
    decrease $breakoutLevel$;
  else
    return S;
  end
until $breakoutLevel = 0$;
return S *and* $weights_C$;

A local search localSearch() is iteratively executed to reach its *local minima* and minimizing its objective function which is represented by the weights of constraints $weight_C$. After each iteration, the weights on the violated constraints are incremented to distinguish the critical constraints from the normal ones. The $breakLevel$ is introduced as the stop criteria since the breakout algorithm still can be trapped in local minima [1]. The $breakLevel$ was set to 10 in our tests as a compromise made between the result quality and the computational time. A little bit different from Eisenberg and Boi Faltings's approach,, we only use the breakout algorithm to detect the constraints which are hard to be satisfied. As mentioned in Theorem 5, this subset hard constraints has the intersection with the IIS. Thus it can be adopted as a preprocessing procedure to reduce the size of the

problem before entering into the IIS detection procedure. The algorithm preprocessing is described as:

Algorithm 2: Preprocessing

input : a set of constraints C
output: a subset hard constraints C_H
/* locate the violated constraints */
$C_H \leftarrow \emptyset, weight_C \leftarrow \emptyset$;
$[C_v, weight_C] \leftarrow breakout(C)$;
$c \leftarrow SelectOne(C_v, weight_C)$;
$C_H \leftarrow C_H \cup \{c\}$;
/* construct the subproblem */
while $Heuristic(C_H)$ *feasible* **do**
  $c \leftarrow SelectFail(Neighbors(C_H, C))$;
  $C_H \leftarrow C_H \cup \{c\}$;
end

First, the breakout identifies the hard constraints and outputs the weights on all constraints. The SelectOne() selects a violated constraint with maximal weight on it. Then the subproblem is constructed by adding the constraints adjacent with hard constraint subset C_H. The next adjacent constraint is chosen by the mean of first-fail [9], whose variables has minimal values consistent with the partial solution found by the heuristic Heuristic() on C_H. The procedure stops when the heuristic() cannot find a consistent solution on C_H. Then the approach continues in extraction step. Since the infeasibility of C_H is not completely proven, it is possible that it will be feasible on extraction step. Thus the algorithm will be restarted with the consistent solution found on C_H.

Algorithm 3: Extract

input : a set of constraints C_H
output: an IS H
repeat
  /* locate the violated constraints */
  $H \leftarrow \emptyset, weight_C \leftarrow \emptyset$;
  $[C_v, weight_C] \leftarrow MinConflict(C_H, \emptyset)$;
  if $C_v = \emptyset$ **then return** C_H *is feasible*;
  /* extend the subproblem */
  repeat
    if $C_v \neq \emptyset$ **then**
      $c \leftarrow SelectOne(C_v, weight_C)$;
    else
      $c \leftarrow SelectOne(C_N, weight_C)$;
    end
    $weight_C \leftarrow \emptyset$;
    $H \leftarrow H \cup \{c\}$;
    $C_N \leftarrow Neighbor(H, C_H)$;
    $[C_v, weight_C] \leftarrow MinConflict(C_N, H)$;
  until $C_v \cap H \neq \emptyset$;
until $Exact(H)$ *is infeasible* ;

In the Extract algorithm, the core engine is the MinConflict() method. The method is derived from the TabCol [24] which is based on the principle of *Minimizing-Conflicts* [25] and working in the reparative manner. The initial solution for TabuCol is randomly generated. Beside the set of constraints of problem as the first parameter, the second parameter of MinConflict() is the subset of critical constraints. In the case of the subset of critical constraints is not empty, the subset of critical constraints should be satisfied in priority. This priority is accented by the following dedicated objective function:

$$f = \delta|H| + |C_r| \quad (1)$$

where $|H|$ is the number of violated constraints inside the critical constraint subset, $|C_r|$ is the number of violated constraints not belonging to the critical constraint subset and δ is a sufficiently great constant to distinguish the critical constraints from the other constraints. During the experiments, the value of δ was set greater than the maximal degree of the variables. The stop criteria of MinConflict() was set to 1000 iterations without improvements. One important side-effect of MinConflict() is the weight for each variable ($weight_C$). For each variable, the weight is incremented by one unit if it was violated during each iteration of search. This precious information is the crucial guidance to identify the critical constraints.

With a closer looking, the Algorithm 3 consists of three procedures - the locating, the extending and the verification. The first procedure consists of identifying a violated constraints subset. Respecting the Theorem 5, this violated constraint subset has an intersection with all IIS constraint subsets inside the given over-constrained problem.

Based on the identified violated constraint subset, one crucial constraint is selected as the first critical constraint probably belonging to an IIS by the method SelectOne(). The selection is based on the weights ($weights_C$) carried out by the MinConflict(). The violated constraint with maximal total weights of its variables is identified as the critical constraint which is potentially belonging to an IIS. The weights statistically indicate the satisfaction difficults of constraints/variables during search [1], the most weighted constraint, the most possibly it is belonging to an IIS.

After locating procedure, the first critical constraint is identified. It's followed by the extending procedure, the Neighbor() enables all the neighborhood constraints which are adjacent with the critical constraint, and prunes all the other constraints from the problem. Then the MinConflict() minimizes the conflicts exclusively among the valid constraints C_N. In case the subproblem constituted by the valid constraints C_N is infeasible, then next critical constraint is located by selecting one violated constraint with maximal weights in the subset of violated constraints of C_N. In case the subproblem is feasible, the constraint with the maximal weights after MinConflict() will be chosen as the critical constraint. This phase extends the subproblem by the means

of the connectivity and identifies the critical constraints until the critical constraints cannot be totally satisfied.

As described, the extending is exclusively applied on the subproblem with the prior satisfaction on the critical constraints. The critical constraints are identified precisely one by one during each iteration while respecting the connectivity rule. With the guidance of the weight heuristic, this technique keeps the IIS smaller and greatly accelerates the extending procedure with focusing on subproblem only.

Since the MinConflict() is a version of Tabu Search [26], [27], which is classified as an incomplete search, it cannot provide a satisfiability prove. Thus the MAC [28] was adopted as a satisfiability testing solver which provides the satisfiability of the critical constraints subset. In case the subset is feasible, all critical constraints are recovered and regarded as normal constraints as the others (Line 2 in Algorithm 3). This *restart* mechanism recovers the critical constraints identified by previous iteration, thus keeps the critical constraints subset from growing extensively. It avoids the MAC verifying the bigger subproblem. As consequence, it may also stop the critical constraints subset extending too earlier and vanish the effects of the extending procedure. Regarding experimental experiences, this restart technique was adopted while the constant extending without restart usually leads to the non-acceptable long computational times and the bigger IS.

When the critical constraints subset is infeasible, it is been proven as an IS. As mentioned in the second part of the Definition 2, if there is no smaller IS inside the identified IS, the IS is an IIS. By concerning the assumption that the weights heuristics [1] always leads to the harder subproblem. Extract was applied on the identified IS, on which, it attempts to find a smaller IS. The iterated execution of Extract stops while there exists no smaller IS. Eventually, the final IS can be considered as an IIS.

The algorithm Extract was described in the constraint-oriented manner, while we are interested in identifying a variable-oriented IIS. For the FAP, the variable-oriented IIS represents the interference zone in telecommunication network. The projection between constraint-oriented and variable-oriented was introduced in [20]. The authors proposed a Saturate procedure to convert all the constraints not identified as critical constraints but exclusively involved with the critical variables as the critical constraints. Thus all constraints among the critical variables are completed.

4. Results and analysis

All the experimental results were carried out on an Intel Core 2 Due E5300 (2.6GHz) machine with 3.2Gb memory. The proposed approach was implemented in C++. The counterpart Abscon(2009) is an updated version of solver proposed by Hemery et al. [18] coded in Java. The two benchmarks used are the real world problem CELAR and academic instances GRAPH (see http://www.inra.fr/mia/T/schiex/Doc/CELAR.shtml). For those feasible instances, the several highest frequencies are pruned to create the infeasible instances.

First we analyse the impact of the local searches embedded in the breakout. The Figure 1 demonstrates the computational times between a simple descent local search and a short limited iteration stop criteria version of TabuCol. Without compromising on the results, the descent local search version is faster than the more sophisticated TabuCol version.

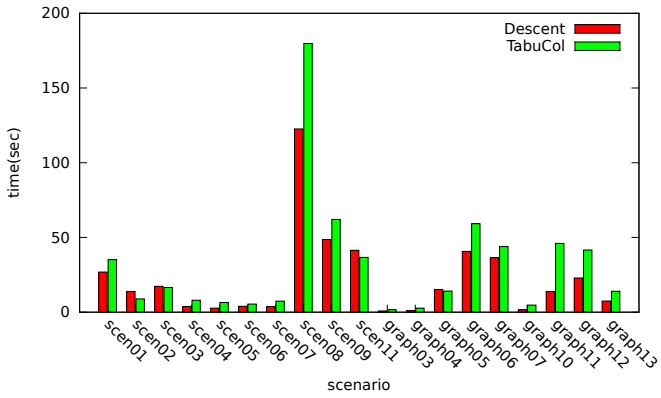


Fig. 1: Descent local search vs. TabuCol

It shows that the effectiveness of the breakout algorithm, even a simple descent local search can provide the precise information of hard constraints. In above figure, the run times are the medians of 50 executions for each instance. The choice is made based on the fact that the medians are more suitable in our case. For example, the distribution of run times of celar01 in CELAR (Figure 2) shows that the majority of run times are just closed to the median as 122.63 seconds.

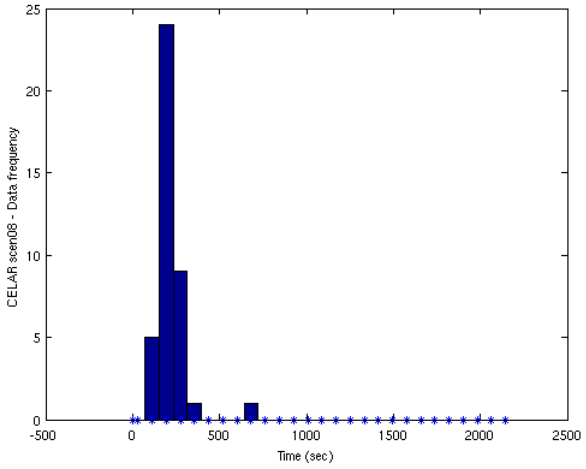


Fig. 2: Distribution of run time on celar08

In Table 1, the first three columns give the description

Scen	$\|V\|$	$\|C\|$	Abscon(2009) $\|V_I\|$	Abscon(2009) $Time(sec)$	Preprocessing + Extract $\|V_I\|$ best(worst)	Preprocessing + Extract $Time(sec)$ median(best)
scen01	916	5548	10	30.33	10(10)	26.86 (10.85)
scen02	200	1235	10	18.02	10(11)	13.89 (3.33)
scen03	400	2760	10	20.47	9(11)	17.28 (7.88)
scen04	680	3967	4	9.87	5(10)	3.77 (2.49)
scen05	400	2598	4	7.81	4(7)	2.66 (1.54)
scen06	200	1322	8	17.91	5(6)	3.91 (1.63)
scen07	400	2865	9	17.22	5(9)	3.74 (2.17)
scen08	916	5744	12	22.07	5(11)	122.63 (14.03)
scen09	680	4103	7	14.97	5(12)	48.6 (6.88)
scen11	680	4103	8	22.88	8(15)	41.41 (13.76)
graph03	200	1134	–	–	2(3)	0.84 (0.36)
graph04	400	2244	4	5.27	2(6)	0.96 (0.83)
graph05	200	1134	8	10.96	4(6)	15.15 (1.77)
graph06	400	2170	5	9.6	4(10)	40.65 (1.92)
graph07	400	2170	5	9.53	4(8)	36.43 (2.14)
graph10	680	3907	–	–	2(4)	1.55 (1.37)
graph11	680	3757	6	11.43	4(8)	13.85 (4.55)
graph12	680	4017	7	12.04	4(7)	22.84 (6.96)
graph13	916	5273	10	20.79	4(6)	7.43 (4.7)

Table 1: Results comparison

of instances, the forth and fifth columns present the IIS's variable size $|V_I|$ and the computational time $Time(sec)$ of Abscon. The last three columns show the experimental results of the proposed Extract algorithm, where $|V_I|$ presents the variable sizes of the IIS, the minimal sizes found and the parenthetical maximal sizes found for 50 runs. Under the $Time(sec)$ column, the median computational time of 50 runs and the minimal computational time in parentheses.

From the results comparison, it shows that the computational times of the Extract algorithm surpass Abscon in several instances. Still the run times are reasonably acceptable (inferior 60 seconds). The relatively longer run times are mainly caused by the combination of MinConflict() heuristic and the chosen weights heuristic. During the locating procedure, it is largely influenced by the initial random solution due to its non-robustness. During the extending procedure, it is easily trapped in the hard subproblem instead of the IS due to its ineffectiveness. Despite its longer computational times, the proposed Preprocessing + Extract still found smaller IIS in most of cases. Among all the tested instances, Preprocessing + Extract only found bigger IIS on one instance comparing to Abscon. For all the runtime surpassed instances, Preprocessing + Extract found smaller IIS. For same size IIS instances, the proposed algorithm always has less computational time. Modeling the IIS detection into hitting set problem demonstrates important impact on the critical constraints identification. Also, the connectivity property guarantees the extension of subproblem towards the right direction and keeps the subproblem smaller.

5. Conclusion and perspectives

In this paper, a hybrid three-procedure algorithm and a pre-processing algorithm were introduced to identify one IIS in the given over-constrained CSP. The relation between the IIS detection and hitting set problem has been clarified. The most importantly, the connectivity property of IIS has been highlighted. Based on these theoretical knowledges, the proposed algorithm is capable to find smaller IIS. And by adopting an hybrid approach, the computational times on large instances have been controlled in an acceptable range. The experimental results demonstrate the competitive performance on the sizes of the identified IIS.

The future works will concentrate on adopting a more effective constructive method during the preprocessing which can precisely locate one infeasible subset.

References

[1] C. Eisenberg and B. Faltings, "Using the Breakout Algorithm to Identify Hard and Unsolvable Subproblems," in *Proceedings of the 9th International Conference on Principles and Practice of Constraint Programming (CP-2003), Lecture Notes in Computer Science*, 2003.

[2] P. Morris, "The breakout method for escaping from local minima," in *Proceeeding of the 11th National Conference on Artificial Intelligence*, Washington, DC, 1993, pp. 40–45.

[3] E. Tsang, *Foundations of constraint satisfaction*. London, UK: Academic Press Limited, 1993.

[4] J. W. Chinneck, "An effective polynomial-time heuristic for the minimum-cardinality IIS set-covering problem," *Ann. Math. Artif. Intell.*, vol. 17, no. 1-2, pp. 127–144, 1996.

[5] K. Aardal, S. van Hoesel, A. Koster, C. Mannino, and A. Sassano, "Models and solution techniques for frequency assignment problems," in *(Updated version of a paper appeared in 4OR 1, 261-317, 2003)*, Jan 2007.

[6] L. Zhang and S. Malik, "Extracting small unsatisfiable cores from unsatisfiable boolean formula," in *Proceedings of Theory and Applications of satisfiability Testing (SAT'03)*, 2003.

[7] R. Bruni and A. Sassano, "Finding minimal unsatisfiable subformulae in satisfiability instances," in *CP*, 2000, pp. 495–499.

[8] R. Bruni, "Approximating minimal unsatisfiable subformulae by means of adaptive core search," *Discrete Applied Mathematics*, vol. 130, no. 2, pp. 85–100, 2003.

[9] B. Smith and S. Grant, "Trying harder to fail first," in *Proceedings of the European Conference on AI (ECAI-98)*, 1998, pp. 249–253.

[10] L. Zhang and S. Malik, "Validating SAT solvers using an independent resolution-based checker: Practical implementations and other applications," in *DATE '03: Proceedings of the conference on Design, Automation and Test in Europe*. Washington, DC, USA: IEEE Computer Society, 2003, p. 10880.

[11] M. Davis, G. Logemann, and D. Loveland, "A machine program for theorem-proving," *Commun. ACM*, vol. 5, no. 7, pp. 394–397, 1962.

[12] I. Lynce and J. Marques-Silva, "On computing minimum unsatisfiable cores," in *International Symposium on Theory and Applications of Satisfiability Testing*, Vancouver,Canada, May 2004.

[13] Y. Oh, M. N. Mneimneh, Z. S. Andraus, K. A. Sakallah, and I. L. Markov, "AMUSE: a minimally-unsatisfiable subformula extractor," in *DAC*, 2004, pp. 518–523.

[14] M. H. Liffiton and K. A. Sakallah, "On finding all minimally unsatisfiable subformulas," in *in International Conf. on Theory and Applications of Satisfiability Testing*. Springer-Verlag, 2005, pp. 173–186.

[15] É. Grégoire, B. Mazure, and C. Piette, "Local-search extraction of MUSes," *Constraints*, vol. 12, no. 3, pp. 325–344, 2007.

[16] P. Galinier and A. Hertz, "Solution techniques for the large set covering problem," *Discrete Applied Mathematics*, vol. 155, no. 3, pp. 312–326, 2007.

[17] M. Liffiton and K. Sakallah, "Algorithms for computing minimal unsatisfiable subsets of constraints," *Journal of Automated Reasoning*, vol. 40, pp. 1–33, 2008. [Online]. Available: http://dx.doi.org/10.1007/s10817-007-9084-z

[18] F. Hemery, C. Lecoutre, L. Sais, and F. Boussemart, "Extracting MUCs from constraint networks," in *ECAI*, 2006, pp. 113–117.

[19] É. Grégoire, B. Mazure, and C. Piette, "On finding minimally unsatisfiable cores of CSPs," *International Journal on Artificial Intelligence Tools*, vol. 17, no. 4, pp. 745–763, 2008.
[20] J. Hu, P. Galinier, and A. Caminada, "On identifying infeasible subsets in constraint satisfaction problems," in *ICAI*, 2010, pp. 615–619.
[21] R. M. Karp, "Reducibility among combinatorial problems," in *50 Years of Integer Programming 1958-2008*, M. Jünger, T. M. Liebling, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, and L. A. Wolsey, Eds. Springer Berlin Heidelberg, 2010, pp. 219–241.
[22] Z. Fu and S. Malik, "On solving the partial MAX-SAT problem," in *SAT*, 2006, pp. 252–265.
[23] C. Ansótegui, M. L. Bonet, and J. Levy, "Solving (weighted) partial MaxSAT through satisfiability testing," in *SAT*, 2009, pp. 427–440.
[24] A. Hertz and D. Werra, "Using tabu search techniques for graph coloring," *Computing*, vol. 39, pp. 345–351, 1987.
[25] S. Minton, M. D. Johnston, A. B. Philips, and P. Laird, "Minimizing conflicts: A heuristic repair method for constraint satisfaction and scheduling problems," *Artificial Intelligence*, vol. 58, no. 1-3, pp. 161–205, 1992. [Online]. Available: citeseer.ist.psu.edu/article/minton92minimizing.html
[26] F. Glover, "Tabu search – part i," *ORSA Journal on Computing*, vol. 1, no. 3, pp. 109–206, 1989.
[27] ——, "Tabu search – part ii," *ORSA Journal on Computing*, vol. 2, no. 1, pp. 4–32, 1990.
[28] D. Sabin and E. Freuder, "Contradicting conventional wisdom in constraint satisfaction," in *Proceedings of ECAI'94*, 1994, pp. 125–129.

Walking Pattern Discrimination based on Wavelet and Fractal Analysis

Yang Xue, Lianwen Jin
School of Electronic & Information Engineering
South China University of Technology
Guangzhou, China

Abstract - *we introduce a simple and effective feature extraction method for human gait classification based on wavelet and fractal analysis of acceleration signal. The focus is on the discrimination between normal walking, fast walking, and step walking. Three types of features are derived from wavelet coefficients obtained by wavelet decomposition of acceleration signals. The wrapper feature subset selection is also used to improve the classification performance. The proposed algorithm can recognize three walking patterns with 98.41% average accuracy. Additionally, we compare our performance with the widely used time-domain and frequency-domain features. Experimental results show that the proposed approach is effective.*

Keywords: accelerometer; walking pattern; discrimination; wavelet; fractal analysis

1 Introduction

Automatic monitoring and recognition of human physical activity with wearable sensors can provide feedback on individual's lifestyle and health status. Such information can help people to maintain their energy balance and stay physically healthy [1]. Recently, promising results from body-fixed accelerometer data for detecting physical activity have been presented [2-4]. The accelerometer output has a good relationship to energy consumption which is widely accepted as the standard reference for physical activity [5-6]. But, for a more precise evaluation, some activities, such as walking, must be further classified into different types to lead the relations between energy consumption and acceleration since this relationship during fast walking may not be the same for normal walking. Because walking is one of the most common and most important daily activities and assessing different walking patterns can provide valuable information regarding energy consumption, some researchers attach great importance to the classification of different walking patterns. For example, flat walking, down slope walking, and up slope walking have been classified in [7], [8], and [9]. However few have tried to investigate the problem of classifying normal walking, fast walking, and step walking, which are also the most common and important daily activities, in greater detail.

Wavelet analysis method is appropriate for the analysis of non-stationary and finite duration signals whose spectral characteristics are changing over time [7, 9]. The statistical characteristics of various walking signals are of finite duration and non-stationary, so wavelet analysis may be a better alternative in such a situation. The acceleration signal during walking represents both the complex pattern and the non-stationary property. The complexity of the signal can be quantified by the fractal dimension [10]. Fractal dimension reveals the deeply embedded correlations (self-similarities) and heterogeneities (dissimilarities) in the signals [10].

In view of this, we propose a wavelet and fractal analysis based feature extraction method for the classification of different walking patterns (fast walking, normal walking, and step walking) in this paper. We first derive three types of features from the decomposed wavelet coefficients of acceleration signals. Then the wrapper feature subset selection is also used to improve the classification performance. Finally, we perform the classification with an SVM and compare our performance with the widely used time-domain and frequency-domain features, which have higher discriminative ability in some literature [1-4].

2 Feature extraction

In this section, we describe the proposed wavelet and fractal feature extraction process in detail.

2.1 Wavelet energy distribution

The 3D accelerations obtained from a 3D accelerometer are $a_x(n), a_y(n), and\ a_z(n)$ .Each component of the tri-axial acceleration $A(n) = [a_x(n), a_y(n), a_z(n)]$ is decomposed *J* levels with the discrete wavelet transform. Wavelet energy distribution (WED) feature is defined as the sum of the squared detail coefficients at level 1-*J* and the sum of the squared approximation coefficients at level *J*. The form of the WED can be given by

$$\begin{aligned} &WED = [E_x\ E_y\ E_z], \\ &E_i = [E_1{}^i E_2{}^i \cdots E_j{}^i \cdots E_J{}^i E_{J+1}{}^i], i \in \{x, y, z\} \\ &E_j{}^i = \sum_k \left(d_{jk}^i\right)^2, j = 1,2,\cdots, J; E_{J+1}^i = \sum_k \left(a_{Jk}^i\right)^2 \end{aligned} \quad (1)$$

where d_{jk} are the detail coefficients in component *i* of the tri-axial acceleration $A(n)$ and a_{jk} are the approximation coefficients in component *i* of $A(n)$; *j* is the scale that represents the dilation index and k represents the index in time, *J* is the depth of the decomposition level.

2.2 Fractal dimension

The most prominent property of the fractal is scale-independent self-similarity. The statistic self-similarity stochastic processes are generally defined as processes whose empirical power spectra are of the form [11]:

$$S_x(\omega) \sim \frac{\sigma_x^2}{|\omega|^\gamma} \quad (2)$$

where $x(n) \in \{a_x(n), a_y(n), a_z(n)\}$, ω is the frequency, σ_x^2 is the variance of the original signal, and γ is the spectral component, which has been proposed to be related to the parameter *H*, $\gamma = 2H + 1$ for fractional Brownian motion and $\gamma = 2H - 1$ for fractional Gaussian noise. *H* is directly related to the fractal dimension (FD) *D* as *D=2-H*.

Recently, the discrete wavelet transform method based on orthogonal wavelet decomposition has been proposed to estimate the fractal dimension D [11], [12]. The variance of d_{jk} at each scale j can be given as follows [11]:

$$\mathrm{var}(d_{jk}) = \sigma^2 2^{-j\gamma} \quad (3)$$

where σ^2 is a positive constant proportional to σ_x^2.

It is, therefore, possible to calculate γ from a simple linear regression of $\log_2(\mathrm{var}(d_{jk}))$ on *j*. $\log_2(\mathrm{var}(d_{jk}))$ is estimated by the variance of wavelet detail coefficients which can be designed by performing a time average of the $|d_{jk}|^2$ at a given scale [12]. After γ has been determined, parameter H and fractal dimension D are estimated

2.3 Wavelet peak

Peaks in the signals of the acceleration can be expected to reveal a great deal more than the basic statistics, such as the minimum, maximum, average of variance over a certain interval [13]. Many previous studies extract peak directly from original time-domain acceleration signal [1- 4]. However, in this paper, we extract peak feature from the approximation coefficients obtained with wavelet transform. In order to detect peaks, we combine three axes acceleration using Eq. (4) to derive a net acceleration $A_{net}(n)$. The net signal $A_{net}(n)$ is decomposed to seven levels using a Daubechies 4 wavelet mother and the peak feature is derived only from the approximation coefficients at level 4. The average height of the peaks and peak number are obtained as the wavelet peak features.

$$A_{net}(n) = \sqrt{\left(a_x\ (n)\right)^2 + \left(a_y\ (n)\right)^2 + \left(a_z\ (n)\right)^2} \quad (4)$$

The wavelet peak and fractal dimension feature are all extracted on a sliding window size of 512 with 50% overlap which has been demonstrated success [14, 15].

Based on wavelet and fractal analysis, three types of features are extracted and then concatenated as a feature vector (Fig. 1).

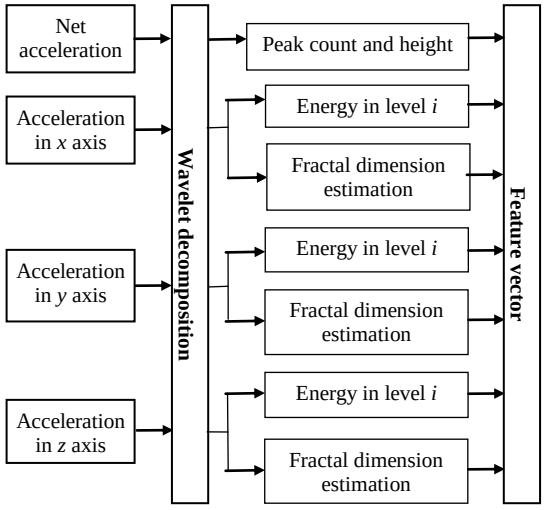


Figure 1. Block diagram of feature extraction

3 Recognition algorithm

To extract three types of features, we first reduce the noise using 1-D Gaussian smoothing and normalize the amplitude for each axis' data. Then, according to Fig. 1, we derive three types of features and concatenate them as a feature set.

For the above extracted feature set, some features may be irrelevant or redundant and not contribute to improve the recognition accuracy. Furthermore, the computational speed

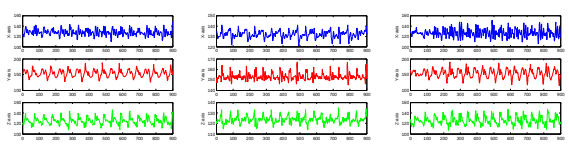
may be slow because of the high dimension of the feature set. Thus, the wrapper feature subset selection (FSS) is taken into account in the classification algorithm.

The wrapper approach is one of the well-known approaches for feature subset selection (FSS) in machine learning [16] and can select effective feature subsets without ignoring the induction algorithm. In the wrapper approach, the FSS is done using the induction algorithm as a black box. The FSS algorithm conducts a search for a good subset using the induction algorithm itself as part of the evaluation function. We chose a best-first search engine that starts with the empty set of features and searches forward. Termination condition is 5 backtracking, which depends on the search engine. The accuracy of the induced classifiers is estimated using accuracy estimation techniques [16]. The accuracy estimation method we used is five-fold cross-validation [17], repeated multiple times with a small penalty (0.1%) for every feature. The number of repetitions is determined by the standard deviation of the accuracy estimate. The induction algorithm used is a Support Vector Machine (SVM) [18] with One-versus-One strategy (OVO). In classification we use the voting strategy of "Max-Wins" to produce the output.

4 Experimental results and discussion

4.1 Experimental data

Accelerometer data are collected using the sampling devices developed by ourselves. Each of these devices contains a tri-axial accelerometer ADXL330, with a dynamic range of ± 3g, which is sampled at 100 Hz. For each subject, data are collected with only one sampling device attached to waist. 42 different subjects (32 male and 10 female) participate in the data collection. Subjects complete a total of three different walking patterns, namely, normal walking (Nwalking), fast walking (Fwalking), and step walking (Swalking). For normal walking, subjects are instructed to perform a gentle walk over a 100m distance, for fast walking to perform a fast walk over the same distance, and for step walking to move the feet alternately in the rhythm of a marching step without advancing for 45 seconds. Fig. 2 shows examples of the raw data and the corresponding colors for the axes (X- blue, Y- red, Z- green).



(a)Nwalking (b) Swalking (c) Fwalking

Figure 2. Examples of raw signals for three walking pattern

4.2 Performance comparison and analysis

To validate the effectiveness of the proposed features, we compare their performance against time-domain features (TF) [1- 4] and frequency-domain features which are also extracted from the raw acceleration signal using a window size of 512 with 256 samples overlapping between consecutive windows. We extracted the first 32 FFT coefficients from each axis of acceleration data and discarded the first coefficients. Preliminary studies [19] showed that the FFT magnitudes have higher discriminative ability than the other features and are therefore used for classification performance comparison. The test is carried out by a leave-one-subject-out cross-validation method in our experiments.

Experiments are first conducted to select Daubechies wavelet mother. There are a large number of wavelet functions from which to select for our fractal dimension estimation. Previous studies [11, 12, 20] have demonstrated that the orthogonal wavelet is used as the best choice to estimate fractal dimension. So, in this paper, we choose Daubechies wavelets.

Table 1 shows the average accuracy of three walking patterns based on different Daubechies wavelets. The features used here contain wavelet energy distribution, fractal dimension, and wavelet peak. The acceleration signals are decomposed into seven levels using Daubechies wavelets. We use seven levels to estimate the fractal dimension since we are unable to estimate γ using the very low frequency scales J, such as 8, 9, and 10 etc. Note that Mallat algorithm for wavelet decomposition is removable algorithm and the decomposed wavelet coefficients will be progressively reduced by half. Therefore, the frequency scale J is not very low.

Table 1 Accuracy based on different DB wavelets (%)

db wavelets	db3	db4	db5	db6	db7	db8
Average	80.16	**94.44**	85.71	84.13	81.75	88.10

From Table 1, the highest classification accuracy is obtained using Daubechies wavelet with the order N=4. So, the acceleration signal is decomposed into seven levels using db4 wavelet mother to estimate the fractal dimension in this paper.

To validate the proposed features (WED+WP+FD), we compare their performance against time-domain features (TF), and FFT coefficients. Table 2 gives the performance comparison of the different features for each of three walking patterns respectively. Our proposed features outperform TF with a 15.07% average accuracy improvement and FFT coefficients with a 4.76% average accuracy improvement.

Table 2 Classification accuracy for the different features (%)

Features	**TF**	**FFT**	**WED+WP+FD**
Dimension	60	465	49
Nwalking	69.05	80.95	85.71
Fwalking	73.81	90.48	100
Swalking	95.24	97.62	97.62
Average	**79.37**	**89.68**	**94.44**

The classification accuracies reported in Table 2 suggest that our proposed features give better classification accuracy than other features.

Finally, Table 3 shows the recognition results using wrapper FSS for different features. The wrapper FSS approach effectively reduces the feature dimensions, for TF from 60 to 6, for FFT from 465 to 12, and for our proposed features from 49 to 14. On the other hand, the wrapper FSS approach significantly improves the recognition performance. The average accuracy for our proposed features is 98.41%, increasing by 3.97% compared with the results without the wrapper FSS step. Obviously, all these results show the superiority of the proposed features, comparing with previous widely used features.

Table 3 Accuracy based on different features with wrapper FSS (%)

Features	TF	FFT	WED+WP+FD
Dimension	6	12	14
Nwalking	95.24	92.86	97.62
Fwalking	76.19	92.86	100
Swalking	95.24	100	97.62
Average	88.89	95.24	**98.41**

5 Conclusions

A simple but yet effective wavelet and fractal analysis based feature extraction method for classification of three walking patterns only using acceleration signals is proposed. Three types of features extracted from the wavelet coefficients are used to distinguish between three different human walking patterns, namely, fast walking, normal walking, and step walking. Wavelet energy distribution features measure signal power. Fractal dimension quantifies the variance progression of the detail coefficient over the different wavelet scales and as such gives a measure of the complexity within the original signal. Wavelet peak features reflect the intensity of the signal. To improve the classification performance, the wrapper feature subset selection is also used. Experiments illustrate the wrapper FSS approach not only effectively reduces the feature dimensions, but also significantly improves the recognition performance. Additionally, we compare the performance of the proposed features with the widely used time-domain and frequency-domain features, all these experimental results show the superiority of the proposed features.

ACKNOWLEDGMENT

This work is supported in part by the National Science Foundation of China (NSFC) (Grant no. U0735004) and GDSFC (Grant no. 2007B010200048, 2008A050200004).

6 References

[1] Ming Li, Viktor Rozgic, and Gautam Thatte, et al. "Multimodal Physical Activity Recognition by Fusing Temporal and Cepstral Information". IEEE Trans. on Neural Systems and Rehabilitation Engineering, vol. 18, No. 4, pp. 1-10, 2010.

[2] M. Ermes, J. Parkka, J. Mantyjarvi, and I. Korhonen. "Detection of Daily Activities and Sports with Wearable Sensors in Controlled and Uncontrolled Conditions". IEEE Trans. on information technology in biomedicine, vol. 12, No. 1, pp. 20-26, 2008.

[3] J.Y. Yang, J.S. Wang, and Y.P. Chen. "Using Acceleration Measurements for Activity Recognition: an Effective Learning Algorithm for Constructing Neural Classifiers". Pattern recognition letters, vol.29, Dec. pp. 2213 – 2220, 2008

[4] N. Ravi, N. Dandekar, P. Mysore, and M. Littman. "Activity Recognition from Accelerometer Data". Proc. of the National Conference on Artificial Intelligence (AAAI 05), pp. 1541-1546, 2005.

[5] C.V. Bouten, K. R. Westerterp, M. Verduin, and J. D. Janssen. "Assessment of Energy Expenditure for Physical Activity using a Triaxial Accelerometer". Med. Sci. Sports Exercise, vol. 26, pp. 1516–1523, 1994.

[6] C. V. C. Bouten, K. T. M. Koekkoek, M. Verduin, R. Kodde, and J. D. Janssen. "A Triaxial Accelerometer and Portable Data Processing Unit for the Assessment of Daily Physical Activity". IEEE Trans. on Biomedical Engineering, vol. 44, pp. 136–147, 1997.

[7] M.N. Nyan, F.E.H. Tay, K.H.W. Seah, Y.Y. Sitoh. "Classification of Gait Patterns in the Time – frequency Domain". Journal of Biomechanics, vol. 39, pp.2647-2656, 2006.

[8] R.K. Ibrahim, E. Ambikairajah, B. Celler, N.H. Lovel, L. Kilmartin. "Gait Patterns Classification using Spectral Features". IET Signals and Systems Conference, pp.98-102, 2008.

[9] N. Wang, E. Ambikairajah, N.H. Lovell, B.G. Celler. "Accelerometry based Classification of Walking Patterns using Time-frequency Analysis". IEEE Int. Conf. on Engineering in Medicine and Biology Society, pp.4899-4902, 2007.

[10] B.B. Mandelbrot. Fractal Geometry of Nature. Freeman Press, San Francisco, 1982.

[11] G. W. Wornell, A. V. Oppenheim. "Estimation of Fractal Signals from Noisy Measurements using Wavelets". IEEE Trans. on Signal Processing, vol.40, pp.611-623, 1992.

[12] P. Abry, D. Veitch. "Wavelet Analysis of Long-range-dependent Traffic". IEEE Trans. on Information Theory, vol. 44, pp. 2-15 , 1998.

[13] K.V. Laerhoven, N. Kern, H.W. Gellersen, B. Schiele. "Towards a Wearable Inertial Sensor Networks". IEE Eurowearable, pp.125-130, 2003.

[14] K.-V. Laerhoven, O. Cakmakci. "What Shall We Teach Our Pants?" The 4th Int. Symposium on Wearable Computers, pp. 77-83, 2000.

[15] R.W. DeVaul, S. Dunn. "Real-time Motion Classification for Wearable Computing Applications". Technical Report of MIT Media Laboratory, 2001.

[16] R. Kohavi, G. H. John. "Wrappers for Feature Subset Selection". Artificial Intelligence, pp.273-324, 1997.

[17] Ron Kohavi. "A Study of Cross-validation and Bootstrap for Accuracy Estimation and Model Selection". Int. Joint Conf. on AI, pp.1137-1143, 1995.

[18] V. Vapnik. The Nature of Statistical Learning Theory. Springer Press, New York, 1999.

[19] S.J. Preece, J.Y. Goulermas, L.P.J. Kenney, D. Howard. "A Comparison of Feature Extraction Methods for the Classification of Dynamic Activities from Accelerometer Data". IEEE Trans. on Biomedical Engineering, vol. 56, pp.871-879, 2009.

[20] Peng Li, Q.B. He, F.R. Kong. "An Approach for Fault Diagnosis of Bearings Using Wavelet-Based Fractal Analysis". IEEE Int. Conf. on Information and Automation, pp.2338-2343, 2010.

Bispectrum Classification of Multi-User Chirp Modulation Signals Using Artificial Intelligent Techniques

Said E. El-Khamy, Hend A. Elsayed and Mohamed M. Rizk
Electrical Engineering Department, University of Alexandria, Alexandria 21544, Egypt.
Emails: elkhamy@ieee.org, hendalielsayed@yahoo.com, mrm_rizk@yahoo.com

Abstract - *Automatic Digital signal type classification (ADSTC) has many important applications in both of the civilian and military domains. Most of the proposed classifiers can only recognize a few types of digital signals. This paper presents a novel technique that deals with the classification of multi-user chirp modulation signals. In this paper, the peak of the bispectrum and its bi-frequencies are proposed as the effective features and different types of classifiers are used. Simulation results show that the proposed technique is able to classify the different types of chirp signals in additive white Gaussian noise (AWGN) channels with high accuracy and the neural network classifier (NN) outperforms other classifiers, namely, maximum likelihood classifier (ML), the k-nearest neighbor classifier (KNN) and the support vector machine classifiers (SVMs).*

Keywords: Multi-User Chirp Modulation Signals, Bispectrum, Artificial Intelligent Techniques, Classification.

1 Introduction

Automatic digital signal type classification plays an important role in various applications. For example, in military applications, it can be employed for electronic surveillance, monitoring; in civil applications, it can be used for spectrum management, network traffic administration, signal confirmation, software radios, and intelligent modems. The early researches were concentrated on analog signals, the recent contributions in the subject focus more on digital types of signals. Primarily, this is due to the increasing usage of such types of signal in novel communication applications. In this paper, we present an automatic digital signal type classifier in additive white Gaussian noise channels for multi-user chirp signals. The used chirp signals are selected such that they all have the same power as well as the same bandwidth. Chirp modulation has been considered for many applications as beacons, aircraft ground data links via satellite repeaters, low rate data transmission in the high frequency (HF) band. It is commonly used in sonar and radar, but has other applications, such as in spread spectrum communications. The linear frequency sweep of a multi-user chirp signals are characterized by the same bandwidth. Each signal is characterized by two different slopes, one slope for each of the two halves of the signal duration. The general expression for these multi-user chirp-modulated (M-CM) signals can be expressed as [1],

$$S_{K1}(t)=\sqrt{\frac{2E}{T}}\cos(\omega_c t+\pi\alpha_K t^2) \qquad 0\le t\le T/2 \tag{1}$$

$$S'_{K1}(t)=\sqrt{\frac{2E}{T}}\cos(\omega_c(t-\frac{T}{2})+K\Delta\omega(t-\frac{T}{2})+\pi\alpha'_K(t-\frac{T}{2})^2) \qquad T/2\le t\le T \tag{2}$$

where, K is the user number, K = 1, 2….. M, M is the total number of users, E is the signal energy in the whole bit duration T, $\omega_c = 2\pi f_c$ is the carrier angular frequency, Δf is the frequency separation between successive users at t=T/2, α_k is the slope within the first half of signal duration, i.e. $0 \le t \le T/2$ and α'_k is the complement slope within the second half of signal duration, **i.**e. $T/2 \le t \le T$. The signal slopes in the two halves of its duration are given by,

$$\alpha_K = \frac{K\Delta f}{T/2} \qquad ,\alpha'_K = \frac{M-K}{T/2}.\Delta f \tag{3}$$

The bandwidth of the different M-CM signals is the same and is given by $B = M\Delta f$ and their time-bandwidth product is given by $\zeta = BT = MT\Delta f$. Automatic digital signal type classifier divided into two main steps is the feature extraction and the classifier. In this classifier, the input signals are passed through an additive white Gaussian noise (AWGN) channels and the received signals are normalized to zero mean and unit variance and the output normalized signal are passed to the feature extraction step. Feature extraction extracts selected peaks of the bispectrum and its bi-frequencies. The classifier uses these features to classify the input signals to get the signal type using different types of Artificial Intelligent Techniques such as maximum likelihood classifier, support vector machine classifier, k nearest neighbor classifier, and neural network classifier. Features extraction based on bispectrum is used to classify mental tasks from EEG signals in [2] and to classify heart rate signals in [3].

This paper is organized as follows. Section 2 Describes bispectrum features extraction. Section 3 describes different types of classifiers. Section 4 shows simulation results. Finally, Section 5 concludes the paper.

2 Bispectrum Features Extraction

The third order cumulant generating function is called the tricorrelation and is shown in equation (4). The Fourier transform of the tricorrelation $C_3^x(k,m)$ is a function of two frequencies and called the bispectrum or the third order polyspectrum in equation (5) [4, 5].

$$C_3^x(k,m) = E[x(n)x(n+m)x(n+k)] \tag{4}$$

$$S_3^x(f_1, f_2) = \sum_{k=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} C_3^x(k,m) e^{-j2\pi f_1 k} e^{-j2\pi f_2 m} \tag{5}$$

The bispectrum or the third order polyspectrum is the easiest to compute and hence the most popular and falls in the category of the Higher Order Spectral Analysis Toolbox (HOSA) [6].

The features are the highest peaks of the bispectrum and the corresponding two frequency components. The selected features are those which show significant differences between the different chirp signals.

3 Classifiers

3.1 Maximum Likelihood Classifier

In the maximum likelihood (ML) approach, the classification is viewed as a multiple hypothesis testing problem, where a hypothesis, H_i, is arbitrarily assigned to the i^{th} modulation type of m possible types. The ML classification is based on the conditional probability density function $p(x|H_i) \quad i = 1,.....,m$, where x is the observation; e.g. a sampled phase component. If the observation sequence X[k], k = 1, . . . , n is independent and identically distributed (i.i.d), the likelihood function (LF), $L(x|H_i)$, can be expressed [7] as

$$p(x|H_i) = \prod_{l=1}^{n} p(X[k]|H_i) \overset{\Delta}{=} L(x|H_i) \tag{6}$$

The ML classifier reports the j^{th} modulation type based on the observation whenever

$$L(x|H_j) > L(x|H_i) \qquad j \neq i; \quad j,i = 1,.....m \tag{7}$$

If the likelihood function is exponential, the log-likelihood function (LLF) can be used due to the monotonicity of the exponent function. Often the expressions of the pdf's are approximate and assume prior information like the symbol rate and SNR. Hence, quasi-optimal rules are defined.

3.2 K-Nearest Neighbor Classifier

K-Nearest Neighbor algorithm KNN) is one of the simplest but widely using machine learning algorithms. An object is classified by the "distance" from its neighbors, with the object being assigned to the class most common among its k distance-nearest neighbors. If k = 1, the algorithm simply becomes nearest neighbor algorithm and the object is classified to the class of its nearest neighbor [8]. Distance is a key word in this algorithm, each object in the space is represented by position vectors in a multidimensional feature space. It is usual to use the Euclidean distance to calculate distance between two vector positions in the multidimensional space. In a two-class classification problem, we would want to choose k to be an odd number to avoid a draw votes. The training process for KNN consists only of storing the feature vectors and class labels of the training samples. One major problem to using this technique is the class with the more frequent training samples would dominate the prediction of the new vector, since them more likely to come up as the neighbor of the new vector due to their large number. K-selection, another important problem we should take into account is how to choose a suitable K for this algorithm, larger values of k reduce the effect of noise on the classification, but make boundaries between classes less distinct. Choosing an appropriate K is essential to make the classification more successful. For two vectors X and Y in an n-dimensional Euclidean space, Euclidean distance is defined as the square root of the sum of difference of the corresponding dimensions of the vector. Mathematically, it is given as

$$D(X,Y) = \left[\sum_{s=1}^{n} (X_s - Y_s)^2 \right]^{1/2} \tag{8}$$

3.3 Support Vector Machine Classifier

SVMs were introduced on the foundation of statistical learning theory. The basic SVM deals with two-class problems; however, with some methods it can be developed for multiclass classification [9]. Binary-SVM performs classification tasks by constructing the optimal separating hyper-plane (OSH). OSH maximizes the margin between the two nearest data points belonging to the two separate classes. Suppose the training set, $(x_i, y_i), i = 1,2,....,l$ can be separated by the $w^T x + b == 0$ where w is the weight vector and b is the bias. If this Hyper-plane maximizes the margin, then:

$$y_i(w^T x_i + b \geq 1, \qquad for\ all\ x_i, \qquad i = 1,2,.....l \tag{9}$$

Those training points, for which the equality in (10) holds, are called support vectors (SV). By using Lagrange multipliers $\alpha_i, i = 1,2,.....,l; \alpha_i \geq 0$, after some computations, the optimal decision function (ODF) is given by

$$f(x) = \mathrm{sgn}(\sum_{i=1}^{l} y_i \alpha_i^* K(x, x_i) + b^*) \quad (10)$$

Where α_i's are optimal Lagrange multipliers. For inputs data with a high noise level, SVM using soft margins can be expressed as follows with the introduction of the non-negative slack variables $\xi_i, i = 1,2,...,l$:

$$y_i(w^T x_i + b) \geq 1 - \xi_i \quad for \quad i = 1,2,...l \quad (11)$$

OSH is achieved by minimizing

$$\Phi = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} \xi_i^k \quad (12)$$

subject to (11), where C is the penalty parameter. In the nonlinearly separable cases, the SVM map the training points, nonlinearly, to a high dimensional feature space using kernel function $K(\vec{x}_i, \vec{x}_j)$, where linear separation may be possible. One of the kernel functions is the Gaussian radial basis function (GRBF) given by:

$$K(x, y) = \exp(-\|x - y\|^2 / 2\sigma^2) \quad (13)$$

where σ is the width of the RBF kernel. After training, the following, the decision function, becomes,

$$f(x) = \mathrm{sgn}(\sum_{i=1}^{l} y_i \alpha_i^* K(x, x_i) + b^*) \quad (14)$$

The performance of SVM depends on penalty parameter (C) and the kernel parameter, which are called hyper-parameters. In this paper we have used the GRBF, because it shows better performance than other kernels. Thus hyper-parameters (σ and C) are selected to have the values one and 10 respectively for all SVMs. There are three widely used methods to extend binary SVMs to multi-class problems. One of them is called the one-against-all (OAA) method. Suppose we have a P-class pattern recognition problem. P independent SVMs are constructed and each of them is trained to separate one class of samples from all others. When testing the system after all the SVMs are trained, a sample is input to all the SVMs. Suppose this sample belongs to class P1. Ideally, only the SVM trained to separate class P1 from the others can have a positive response. Another method is called the one-against-one (OAO) method. For a P-class problem, $\frac{P(P-1)}{2}$ SVMs are constructed and each of them is trained to separate one class from another class. Again, the decision of a testing sample is based on the voting result of these SVMs.

The third method is called a hierarchical method. In this method the received signal is fed to the first SVM (SVM1). SVM1 determines to which group the received signal belongs. This process will be continued in the same manner until the signal types are identified by the last SVMs. One of the advantages of this structure is that the number of SVMs is less than in cases of OAO and OAA.

3.4 Neural Network Classifier

We have used a MLP neural network with back-propagation (BP) learning algorithm as the classifier. A MLP feed forward neural network consists of an input layer of source nodes, one hidden layer of computation nodes (neurons) and an output layer. The number of nodes in the input and the output layers depend on the number of input and output variables, respectively and the number of nodes in the hidden layer is 17 neurons. And the classifier is allowed to run up to 5000 training and with MSE is taken to be 10-6, the activation functions used for hidden layer and for output layer respectively are Hyperbolic tangent sigmoid and Linear transfer function [10].

4 Simulation Results

In this section, we evaluate the performance of automatic digital signal type classifier of the of the considered multi-user chirp modulation signals. The used eight chirp signals (S1, S2, S3, S4, S5, S6, S7, S8) are generated using equations (1) and (2) by putting M=8. Assume T=1 sec, $f_c = 1kHz$ and the time-bandwidth product $\zeta = 1500$. Plots of the instantaneous frequencies of these eight chirp signals are shown in Fig. 1. Each signal is chosen to have 150 realizations and 4096 samples (1 second), they are then divided into 100 realizations for training and 50 realizations for testing data sets. White Gaussian noise is added to these signals and features are extracted. The features are extracted by first divide each signal into segments in our study the length of the segment (Ns) is 32 samples and apply the function (bispeci) from the Higher Order Spectral Analysis Toolbox in [6] to estimate the bispectrum using the indirect method where maximum number of lags are 31 and without overlapping and biased estimate. Then features are extracted by taking the maximum peaks of the absolute value of the bispectrum and the corresponding two frequencies of that peaks are extracted. Fig. 2 shows the contour plot of the magnitude of the bispectrum of the signal S2 for Ns=32 and the regions R1 and R2. The number of peaks is high so it needs to reduce. First, the region of the bispectrum (R1) is used. We note from our study that, there is symmetry so we use only the region R2. The features values are computed under the constraints of zero mean, unit variance and noise free where f11 and f21 are the value of the frequency of the first high peak in the horizontal and vertical axes respectively and P1 is the value of that peak. Also f12 and f22 are the second high peak in the horizontal and vertical axes respectively and P2 is the value of that peak. All these six features are called F3 and used for classification. This method is compared with using the features F2 (P1 and P2) and features F1 (f11, f21 and P1). If we divide each signal into segments with length (Ns) of the 128 samples, the mesh plot of the magnitude of the bispectrum of the signal S2 for Ns=32 and 128 are shown in Fig. 3 and Fig. 4. Tables 1 and 2 show the features for the eight chirp signals when Ns are 32 and 128 samples there six features are called F4. Fig. 5 shows the contour plot of the eight signals and the region used R2 for

Ns=32. The performance of the mutli-user chirp modulation signals using Maximum Likelihood Classifier using features F1, F2, F3, and F4 are shown in Fig. 6. Fig. 7 shows the performance using K nearest neighbor Classifier for k=1 and using F1, F2, F3, and F4 and the performance using different K nearest neighbor Classifier for k=1, 3, and 7 using F4 is shown in Fig. 8. Fig. 9-11 show the performance using the one-against-all (OAA), one- against-one (OAO), and Hierarchical support vector machine classifier using F1, F2, F3, and F4. Fig. 12 shows the performance of the one-against-all (OAA), one- against-one (OAO), and Hierarchical support vector machine classifier using F4. Fig. 13 shows the performance using multilayer perceptron neural network using features F1, F2, F3, and F4. From the results we show that using F4 as features outperforms using F1, F2, and F3. Fig. 14 shows the performance of multi-user chirp modulation signals using F4 features and different classifiers. From these results, we note that the MLP classifier is the best classifier.

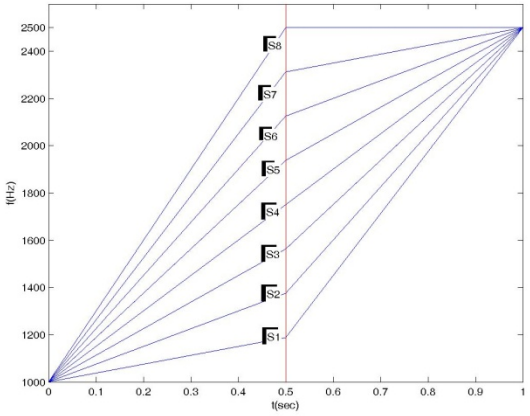


Fig. 1. Instantaneous frequency of Multi-User Chirp modulation signals over the carrier frequency.

TABLE 1
THE FEATURES FOR EIGHT MULTI-USER CHIRP MODULATION SIGNALS FOR SEGMENT LENGTH 32 SAMPLES

	S1	S2	S3	S4	S5	S6	S7	S8
f11	0.1613	0.1613	0.1774	0.1613	0.1613	0.1613	0.1613	0.1613
f21	0.1613	0.1613	0.1774	0.1613	0.1613	0.1613	0.1613	0.1613
P1	5.7133	10.604	2.1705	4.3127	10.042	3.8803	1.8814	6.8151
f12	0.1613	0.1774	0.1774	0.1774	0.1613	0.1774	0.1613	0.1613
f22	0.1452	0.1613	0.1290	0.1613	0.1452	0.1452	0.1452	0.1452
P2	2.2340	3.2359	2.1705	1.8729	2.9447	1.9244	1.3337	2.2120

TABLE 2
THE FEATURES FOR EIGHT MULTI-USER CHIRP MODULATION SIGNALS FOR SEGMENT LENGTH 128 SAMPLES

	S1	S2	S3	S4	S5	S6	S7	S8
f11	0.1654	0.1654	0.1654	0.1654	0.1654	0.1614	0.1654	0.1654
f21	0.1614	0.1654	0.1654	0.1654	0.1654	0.1614	0.1654	0.1654
P1	15.674	170.53	156.56	58.556	44.972	26.605	99.307	10.144
f12	0.1693	0.1693	0.1654	0.1693	0.1614	0.1732	0.1693	0.1654
f22	0.1614	0.1614	0.1614	0.1693	0.1614	0.1614	0.1614	0.1624
P2	15.674	72.161	63.492	32.705	40.507	26.605	49.269	9.0932

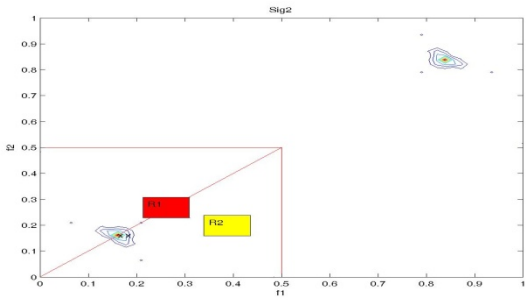


Fig.2. Contour plot of the magnitude of the bispectrum of the signal S2 for Ns=32 on the bi-frequencies (f1, f2) and the regions R1 and R2.

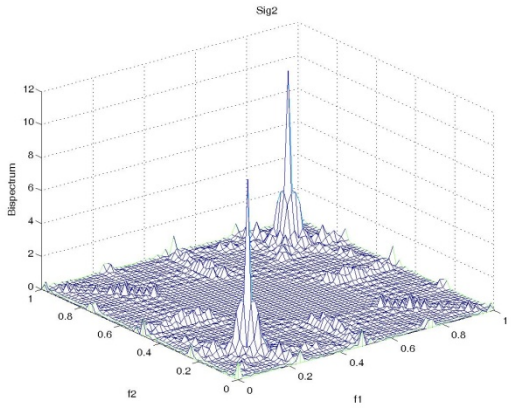


Fig. 3. Mesh plot of the magnitude of the bispectrum of the signal S2 Ns=32

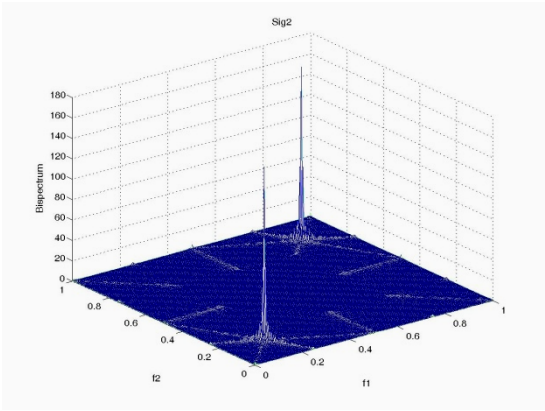


Fig. 4. Mesh plot of the magnitude of the bispectrum of the signal S2 Ns=128

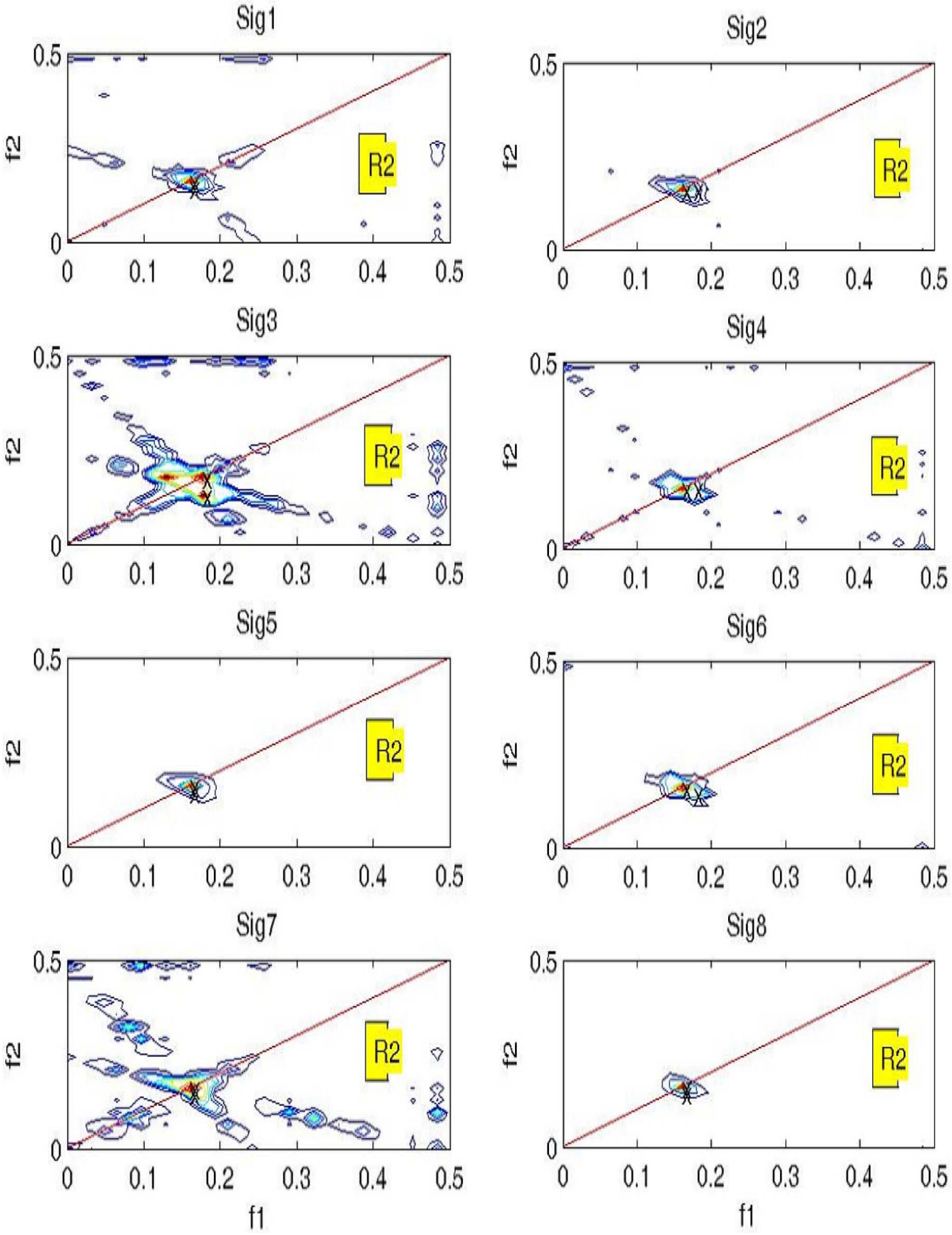


Fig. 5. Contour plot of the magnitude of the bispectrum of the eight signals on the bi-frequencies (f1, f2) and the region R2.

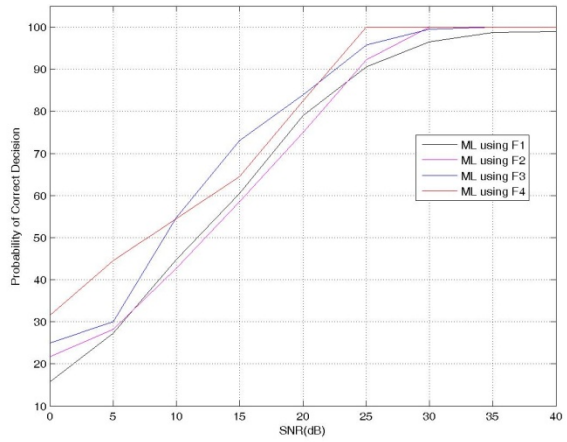


Fig. 6. The performance of multi-user chirp modulation signals using ML Classifier.

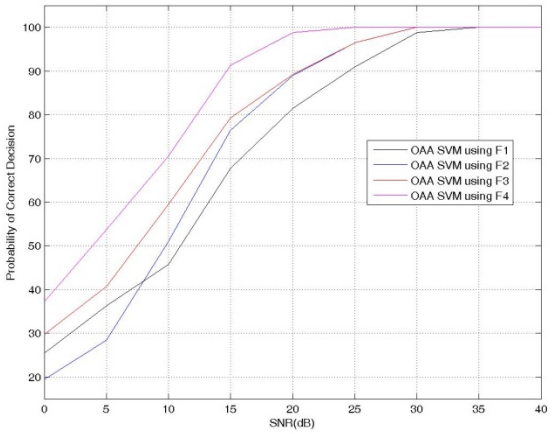


Fig. 9. The performance of multi-user chirp modulation signals using OAA SVM Classifier.

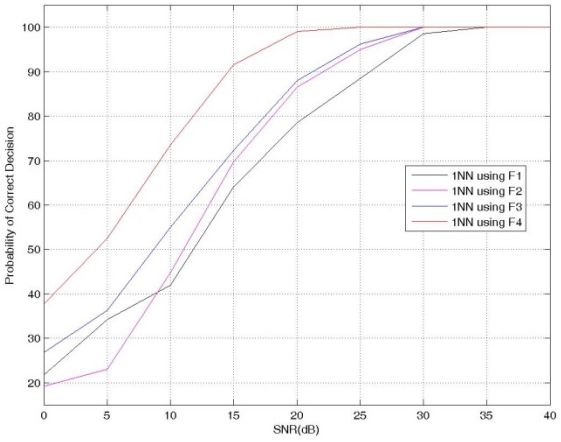


Fig. 7. The performance of the multi-user chirp modulation signals using 1NN Classifier.

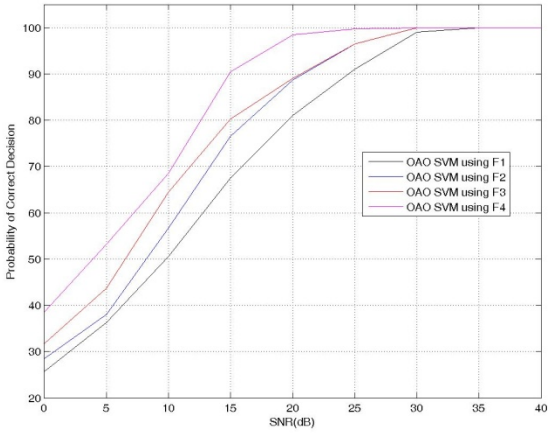


Fig. 10. The performance of multi-user chirp modulation signals using OAO SVM Classifier.

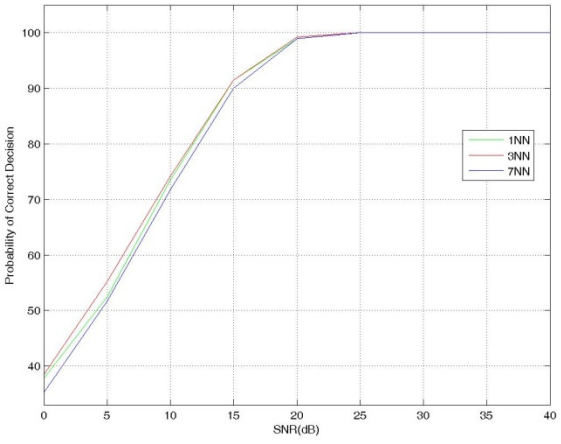


Fig. 8. The performance of the multi-user chirp modulation signals using KNN Classifier and K=1, 3, and 7.

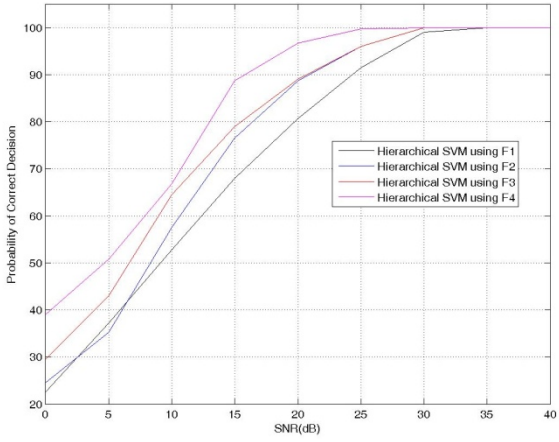


Fig. 11. The performance of multi-user chirp modulation signals using Hierarchical SVM Classifier.

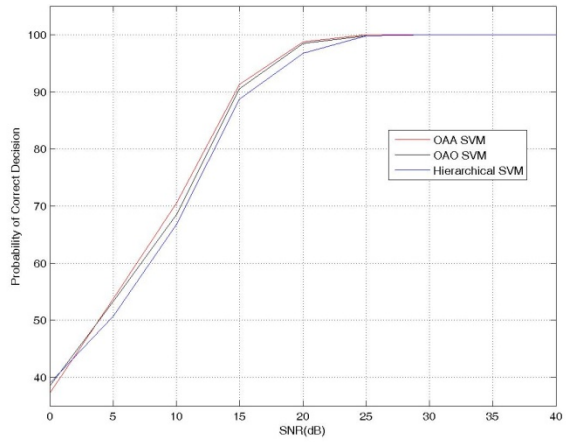


Fig. 12. The performance of multi-user chirp modulation signals using different SVM Classifiers.

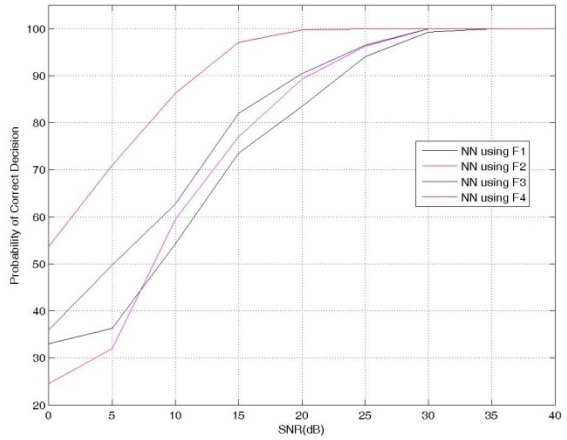


Fig. 13. The performance of the multi-user chirp modulation signals using MLP Classifier.

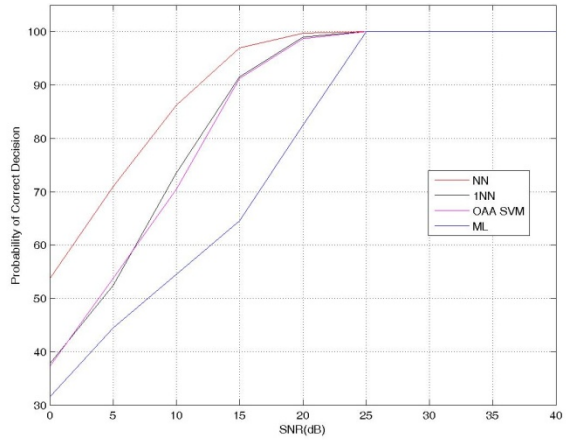


Fig. 14. The performance of the multi-user chirp modulation signals using different Classifiers.

5 Conclusions

In this paper, we presented multi-user chirp modulation signals classification using bispectrum. In this method, different types of classifiers are used. We have showed the dependence of the classifier performance on the classifier parameters and the features used and length of each segment. Simulation results show that the performance of the multilayer perceptron neural network classifier is better than other classifiers such as maximum likelihood classifier, k nearest neighbor classifier, and support vector machine classifiers and the performance of using Features F4 is better than using features F1, F2, and F3. In the support vector machine classifiers, the performance of the one-against-all classifier is better than one-against-one and hierarchical support vector machine.

6 References

[1] S. E. El-Khamy, S. E. Shaaban and E. A. Thabet, "Multi-user chirp modulation signals (M-CM) for efficient multiple access communication systems," in proceedings of the thirteenth national radio science conference, March 19-21, 1996, Cairo, Egypt.

[2] S. M. Zhou, J. Q. Gan, and F. Sepulveda, "Classifying mental tasks based on features of higher-order statistics from EEG signals in brain–computer interface," Information Sciences 178 (2008), pp. 1629–1640.

[3] C. K. Chua, V. Chandran, R. U. Acharya and L. C. Min , " Cardiac Health Diagnosis Using Higher Order Spectra and Support Vector Machine," The Open Medical Informatics Journal, 2009, 3, pp. 1-8.

[4] C. L. Nikias and J. M. Mendel, "Signal Processing with Higher Order Spectra," IEEE Signal Processing Magazine. July 1993.

[5] C. L. Nikias and A. P. Petropulu, "Higher-Order Spectra Analysis," A Nonlinear Signal Processing Framework, PTR Prentice-Hall, Englewood Cliffs, NJ, 1993.

[6] High-order spectral analysis Toolbox (2011, February 9) [Online].Available: http://www.mathworks.com/matlabcentral/fileexchange/3013.

[7] A. V. Rosti, "Statistical Methods in Modulation Classification," Master of Science Thesis, Tampere University of Technology, department of information technology, June 1, 1999.

[8] H. Yoshioka, "A fast modulation recognition technique using nearest neighbor rules with optimized threshold for modulation classification in Rayleigh fading channels," Proc. WPMC, (2002), pp. 1049-1052.

[9] C. Burges, "A tutorial on support vector machines for pattern recognition," Data Mining and Knowledge Discovery, 2 (1998), pp. 121–167.

[10] A. Ebrahimzadeh, M. Ebrahimzadeh, "An Expert System for Digital Signal Type Classification," Journal of Electrical Engineering, vol. 58, NO. 6, 2007, pp. 334–341.

Using Visual Fingerprints of Places for Robotic Localization

B. J. Wimpey[1] **and W. D. Potter**[1,2]
[1]Computer Science Department, University of Georgia, Athens, Georgia, USA
[2]Institute for Artificial Intelligence, University of Georgia, Athens, Georgia, USA

Abstract— *We describe a framework to enable an autonomous mobile robot to recognize locations, with applications to topological mapping and loop closing. Our system encapsulates various visual feature "ridges" into location "fingerprints" which can be stored as nodes in a topological map or used to compare with the map for localization. One of the ridge features we use is SIFT, the Scale Invariant Feature Transform, which adds to the system the ability to recognize interesting locations with its keypoints. SIFT keypoint extraction performs rather poorly, however, in areas of low contrast. Therefore, we complemented it with other machine vision techniques including object recognition, object detection, straight line statistics, color histograms and text recognition. We call the system SPLINTR, Spatial PLace recognition IN a Topologically mapping Robot. Here we describe the SPLINTR system as a whole and the visual feature ridges; experimental results are presented showing SPLINTR's use in robotic localization.*

Keywords: robotics, machine vision, localization

1. Introduction

As we approach the location recognition problem, we must consider several questions. What is required for an autonomous mobile robot to be able to recognize a location it has visited before? Furthermore, what features are deemed important enough for the robot to remember as it navigates and explores, and can we tie such features together to help strengthen the approach?

One visual technique used in the localization and mapping domain is to extract keypoint features from the environment [1], [2]. SIFT [3], the Scale Invariant Feature Transform, is one such method of keypoint extraction. SIFT features are invariant to scale and orientation changes and can handle some changes in light intensity and 3D viewpoint location as well. SIFT features have weaknesses, however, especially if the image contains large amounts of low contrast areas. For interesting features to exist there needs to be texture and non-uniformity. Thus if we have other layers within our system to provide support when a single technique like this might fail, we produce a more robust, reliable and accurate system. Therefore, we designed a system with multiple layers, or "ridges," so that the robot does not have to depend on one single visual feature in order to map and recognize locations. Similar to how the patterns of friction ridges on the fingertips form the characterizing nature of human fingerprints [4], combining the results of these feature "ridges" forms our idea of a location "fingerprint," facilitating place recognition; when these fingerprint nodes are associated together between locations, the robot is able to topologically map its environment. The multiple ridges for this system include SIFT keypoints, object detection and recognition, line detection, color histograms, and text recognition. Our system, called SPLINTR, Spatial PLace recognition IN a Topologically mapping Robot, is an attempt at combining mobile robots, Artificial Intelligence and machine vision techniques for use in recognizing a location and mapping. What is presented is a meshing of somewhat simple machine vision techniques combined with more sophisticated approaches. Combining the ridges as a single node encapsulates the signature of the location and allows it to be stored as a single entity. Each ridge in a fingerprint node is one of our set of machine vision results after being applied to the scene image (or images) acquired at each location visited. Given our complementary, multilayered approach, the system should work even when there may be few landmark type features to work with.

2. Related Work

There has been work in the area of location recognition using simple image features. In [5], Lamon et al. used a robot to localize itself using location "fingerprints" represented as strings of characters. Each character in the string fingerprint represented a detected color patch or a vertical straight line edge. For each feature detected in the image, its representative character was recorded in the string fingerprint in the order it was detected at that location. Locations within the map were also represented as strings. The robot would then try to localize by string-matching the resulting character-based fingerprint of the current location to one in the map. In [6], Tapus and Siegwart extended the string fingerprint to also include corners detected from laser scans. This time the string fingerprints were used for topological mapping (as opposed to the localization method of [5]); the location fingerprints were used to either update a mean fingerprint of a node or to add a new node to the map should successive fingerprints be sufficiently different. Cummins and Newman [7] used SIFT features to represent locations in their work on appearance-based localization and mapping. Images of locations with distinctive features are needed for their function to output a high probability that the images came from the same location as one in their map. SIFT

features were used by Martinez-Gomez et al. [8] for their place recognition approach in the RobotVision@ImageCLEF competition [9], in which participants were to classify query images into the place categories from which the images were taken. Fraundorfer et. al [10] combined SIFT features with viewpoint invariant features, or VIP features, for their place recognition technique. In a kind of combination between localization and image retrieval, Hays and Efros [11] provided a localization technique based on how similar a query image is to their database of GPS-tagged images. Their IM2GPS system [11] involved matching scenes of images with a huge GPS-tagged image database (6 million images), and they used various features including straight lines, gist [12], CIE $L^*a^*b^*$ color histograms, texton histograms [13], and small, 5x5 color (CIE $L^*a^*b^*$) images. In Xing and Pronobis [14], the authors also make use of global and keypoint image features for place recognition, though their choice of global image features are different, and they do not use features such as object detection, object recognition, or text recognition.

While localization and mapping work has employed various image features, and has used SIFT features and object recognition [15], our work utilizes a novel combination of complex visual techniques (SIFT, object recognition, object detection and text recognition) in conjunction with more general, global image features (color histograms and statistics of straight lines).

3. The Multilayered Fingerprint Approach of SPLINTR

Within the multilayered fingerprints of the SPLINTR system, each ridge helps to fill a niche in capturing the signature of a location. The SIFT keypoints, for example, provide recognition of details in the environment, including characteristic patches of texture and interesting features. SIFT keypoints can also be used for object recognition. Using a database of SIFT keypoint sets extracted from object models, those SIFT points can be matched with SIFT points from the current scene; if enough matches are found, we can consider the object to be present in the scene image. Therefore, while exploring, detection of SIFT keypoints allows the robot to distinguish landmark objects as well as other texture-based features present within the environment.

Other ridges were chosen to help the robot to succeed not only in areas with many SIFT features, but also in locations with few distinguishing keypoint features – areas of the environment in which the dominant characteristics are more general in nature. Accumulating statistical measures of straight line segments, including their lengths, orientations and color, gives the system more representative characteristics to assign to each location's fingerprint. Color histograms are used as one of the fingerprint ridges in order to capture a color distribution snapshot of the location. Color histograms alone may be too general to match to specific locations, but they can be useful to thin the list of possible locations, as was the case with the localization method used by Blaer and Allan [16]. Straight line information and color histograms can also be used to corroborate evidence between the current scene and fingerprints in the map. Furthermore, recognition of text is highly useful within man-made environments, as textual characteristics are likely to be found within buildings to mark offices and even outside to represent the landmarks of street signs, house numbers or business signs.

The world model for the SPLINTR fingerprint localization and mapping system consists of the topological map of fingerprints, each containing the signature of a discrete location from the robot's exploration. To determine if the robot is currently at a location within its map, the robot can compare each ridge of the fingerprint extracted from its current location to those fingerprints within its map. Comparison measures between ridges are discussed in Section 4. Should the similarity between a node in the map and the current location's node be above a certain threshold, it could then be considered the correct current location.

4. Implementation of the Fingerprint Ridges

Next we describe the ridges developed for the SPLINTR fingerprint mapping system. As more features are used to record a location, the robot gains more evidence to match between its scene and the fingerprints in its map, making it less dependent on a single fingerprint ridge to make a decision.

4.1 SIFT Keypoints and Object Recognition

Lowe's SIFT keypoints [3], as mentioned earlier, are scale and orientation invariant, while being partially invariant to affine changes and illumination differences. SIFT features lend themselves to object recognition as well, as described by Lowe [3]. Therefore, not only can environmental locations be described with our system using SIFT points, they can further be characterized based on the objects recognized from those SIFT points.

During SIFT ridge formation, we extract SIFT features from the scene image, save those features, and also decide if there are any model objects within the scene image by matching the scene features with the SIFT features of our model database. The vision system of SPLINTR uses an open source SIFT implementation from Rob Hess [17] as well as Hess's open source implementation of the SIFT descriptor matching algorithm as described by Lowe [3]. Thus we can extract SIFT features from the scene image, store them within the SIFT ridge, and extract nearest neighbor matches using the kd-tree implementation of [17].

The object recognition cycle first starts with a database of objects to find in the environment. Each object's model

Fig. 1: (Left, Center) SIFT Features on two sample objects. The arrows indicate the keypoint's orientation and scale at which it was found. (Right) Example of object recognition using the two sample object models, with the top of the object indicated with a white line.

has its associated SIFT keypoints representing the object. For each feature in the scene image, one must find its closest match amongst the SIFT features of the model image (repeated for each model in our database). Comparing the descriptor's distance to the closest match with the distance to the second closest match, the keypoint should be much closer to the closest match than the second closest match. If they do not meet a certain threshold, the match is discarded.

SIFT keypoints, when extracted, are assigned an orientation, which helps facilitate the rotation invariance property of SIFT. The orientation is a representation of a popular gradient direction in the neighborhood around the SIFT keypoint. If we are matching SIFT points of a particular object, and the object is rotated in the scene image, then the SIFT points of the object will therefore have a common relative rotation difference to the SIFT points of our object model. To exploit this property, we create a histogram of the difference of orientations of the matching SIFT points between our scene and our object. Such a technique is described by Lowe [3] as a Hough transform voting mechanism. Thus the most popular difference of orientation of matched keypoints defines the orientation angle of the object in the scene. We can use this value to eliminate false matches, that is, any keypoints with too great a difference in orientation to the consensus orientation for the model's match in the scene. The orientation difference is divided into 24 bins, and votes are cast into the two nearest bins. The features also vote for the location of the object within the scene image based on that feature's projection into the image (based on the match's scale and orientation). To vote for object position, the image area is divided into 80 cells, 10 cells wide (eight within the valid area of the image and one extra cell range on each side) and eight tall (six cells on the image and one above and below the image bounds), with the cells on the outside of the image to catch projection votes outside the image bounds.

We use OpenCV to create a homography between the matching SIFT points, and we use OpenCV's perspective transform to put a bounding box around the recognized object in the image, as illustrated in Fig. 1. In the current implementation, we require a minimum of five matches to call OpenCV to find a planar homography for the corresponding SIFT points.

The SIFT similarity measure between the current scene image and the SIFT ridge in the map node is a ratio of good matches to the total matches, with a "good" match being within a 36 degree orientation difference window, 18 degrees above and 18 degrees below a 0 orientation difference.

4.2 Objects Ridge

The Objects Ridge consists of both objects recognized using SIFT and objects detected. To perform object detection, we used OpenCV to create multiple training samples to be used for object detection training. If multiple images of a single object type were used for training, we used Seo's MergeVec[18] to merge the individual training sample results into a single file accepted by OpenCV. Furthermore, we used OpenCV's cascade classifier training utility to create boosted classifiers of objects (particularly, using the Gentle AdaBoost boosting algorithm), and OpenCV's multiscale cascade classifier object detection method. When provided with a trained classifier, OpenCV can then return the locations of the detected object; this process is repeated for each trained classifier of objects we wish to detect. Doors are detected using the door candidate detection method of J. Tarver [19], tweaked slightly using some of the edge enhancement techniques from [20]. Objects recognized using SIFT are passed to the objects ridge to be merged with objects detected. For object ridge formation and to facilitate comparing the recognized object ridges between two fingerprints, the recognized and detected objects are first represented as a string of characters using an acronym. The acronym of objects recognized and detected is a list of object names shortened to a string of just their first characters in the object name, in the left to right order they were recognized in the image. For example, given two objects detected that are "door" and "firealarm," the representative objects ridge string representation will be reduced to the first letters of each object name in the order detected, "df."

Next we describe the similarity measure for comparing object ridges. The acronym strings of objects recognized and detected from the current scene image and the map node are compared. Our object list similarity measure is $(numMatches + (0.5 * theMaxSeq))/(1.5 * maxLength)$, where $numMatches$ is the number of objects recognized that the two strings have in common, and $maxLength$ is the longest string between the two nodes. $theMaxSeq$ is the longest subsequence the two strings have in common (that is, the number of objects in the right order, but not necessarily next to each other); this is calculated using the longest increasing subsequence algorithm, which we calculate using a dynamic programming solution [21], [22].

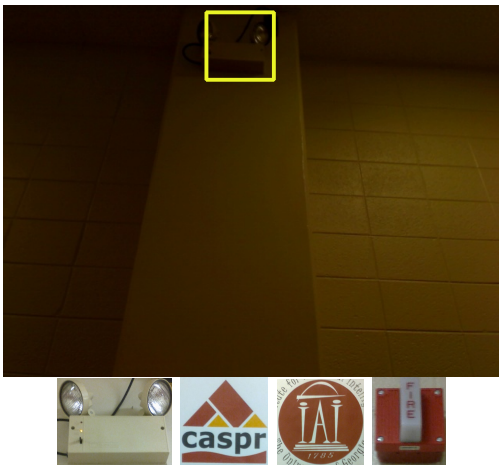


Fig. 2: Example of using OpenCV object detection, showing the result of detecting the emergency light. In our experimentation, we used boosted classifiers to detect emergency lights, the CASPR logo, the IAI logo, and fire alarms.

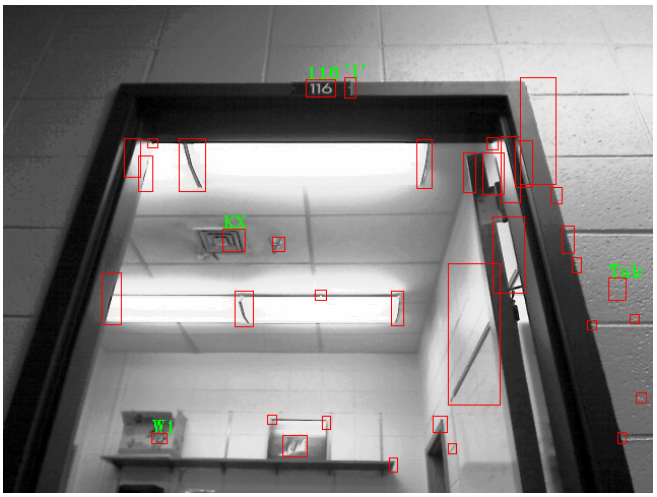


Fig. 3: Results of text detection, as indicated by the red boxed regions, and the results when passing that area of the higher resolution image to the OCR engine.

4.3 Straight Lines Ridge

Lines are common in man-made structures, and can be plentiful in hallways with doors and door frames, hallway archways, and grooves in the walls resulting from stacked bricks or cement blocks. To acquire the lines ridge, OpenCV is used to first convert incoming images to grayscale and then apply the Canny operator. An OpenCV version of the Hough transform is then applied to the resulting Canny filtered image. The straight line ridges are represented as three normalized OpenCV histograms of statistics collected from the straight lines extracted. Length and orientation of straight lines make up two histograms, as used in [11]; the third histogram is a 2D color histogram of the hue-saturation pixel values of the straight lines.

Since line ridges are composed of three separate histograms, to compare line ridges, we simply compare the associated statistics collected in the histograms using OpenCV's correlation comparison measure. In OpenCV, the correlation comparison measure takes in two histograms and returns the distance between the two histograms. The OpenCV correlation between histograms H_1 and H_2 is computed as:

$$d(H_1, H_2) = \frac{\sum_i (H_1[i] - \overline{H_1})(H_2[i] - \overline{H_2})}{\sqrt{\sum_i (H_1[i] - \overline{H_1})^2 (H_2[i] - \overline{H_2})^2}}$$

where $\overline{H_k}$ represents the mean of H_k's histogram values. The orientation, length and color histogram correlation results are averaged together to form the overall lines ridge similarity measure.

4.4 Color histograms

If the robot is in a location with generic features, one way to still capture the location's fingerprint is to make a color histogram of the location. Thus even if the robot is not capable of extracting useful SIFT features due to the fact that it is surrounded by uniform colored objects like walls, it might still extract useful information, given the walls may be made of a certain color scheme, such as a common off-white painted color or the redness of bricks. A simple technique such as a color histogram can create a snapshot of the color distribution of the location of the robot. This adds to the ability to distinguish or verify areas with many SIFT keypoint features as well, since matching SIFT points in a drastically different color distribution is a red flag that this may not be the same location as the node in the map. We use hue-saturation histograms to represent the color distribution of the image; OpenCV is used to convert the input image from the RGB color space into an image in the HSV color model. A two dimensional histogram is built from the hue and saturation channels of the HSV image, with 30 hue bins and 32 saturation bins. To compare the color histogram ridges of two location fingerprints, we apply the OpenCV histogram correlation measure.

4.5 Multi-resolution Text Recognition

In our implementation of the text recognition ridge of the SPLINTR fingerprint system, first a text detection algorithm [20] is applied to the input image. Possible areas of text within the image are returned by the detection algorithm and bounded to be further processed and analyzed. After text detection, that area of the image is analyzed in the larger resolution image. Each input image to the system is a smaller resolution version, used for feature extraction, and the larger resolution image is used to gain better pixel information to be used in text recognition. Currently, the sub-images are thresholded into a binary image, and components are analyzed using an open source OpenCV library called

cvBlob [23] and small components are removed. After preprocessing, the results of the sub-images are passed to the OCR engine for recognition. For optical character recognition, we use the Tesseract OCR engine [24], a Google code open source project under the Apache License. We currently employ Tesseract OCR version 2.04. After a text detected sub-image is preprocessed by binarization and noise removal, the data array containing the pixel intensities is fed into the Tesseract OCR engine, which processes the sub-image and returns text that passed its recognition engine. To measure the similarity between text ridges, we compare each text string in the input fingerprint text ridge with all strings in the map fingerprint to find its best string match; to measure the distance between strings we use the Levenshtein distance [25]. The best matches' distances are summed, and the maximum distances are summed (that is, the maximum distance this string could be from its best match). Our final text similarity measure is:

$$textSim = 1.0 - (minDistanceSum / maxDistanceSum)$$

4.6 Overall Fingerprint Similarity

Currently, the overall similarity match between fingerprints is formed from equally weighting the individual ridges to form a single similarity measure:

$$totalSim = w * siftSim + w * objSim + w * linesSim + w * colorhistSim + w * textSim$$

and the weight, w, is calculated as $w = 1.0/NumberOfRidges$. If no lines are extracted from either fingerprint being compared, then that ridge does not contribute to the overall similarity, and the individual weights of the remaining ridges are increased equally. Similarly, if no objects are detected or recognized in either fingerprint being compared, then the objects ridge, likewise, does not contribute to the overall similarity measure.

5. Experimental Results

We experimented using the SPLINTR system for use in the task of localization. We chose the testing environment to be a portion of the first floor of the Boyd Graduate Studies Research Center building on the campus of the University of Georgia. We collected images from 46 discrete locations approximately one meter apart in the environment to form the robot's "map." In order to collect images that also captured text information such as door numbers or name plates, the camera was extended above the ER1 robot chassis approximately 3 feet and tilted upward approximately 45 degrees. The environmental image database consists of images from the robot's left and right perspective, which are fused into a single image as illustrated in Fig. 6c. Furthermore, larger resolution images were fed into the SPLINTR system to facilitate multi-resolution text recognition, as described earlier in Section 4.5. To form a set of query images to test for localization, a test set was collected by moving the robot to new locations and acquiring images. Map nodes are shown as black dots, and the test nodes are shown as stars, as seen in Fig. 4. The test nodes and the nearest map nodes are labeled in the figure. To form the map, the system reads in each environment image and extracts the visual feature ridges. Once the ridge features are extracted, they are stored as a fingerprint node within the map; this process continues until all environment images are processed and their respective fingerprint nodes are formed. To test for localization, a test image is inputted to the system, its ridge features are extracted to form the fingerprint, and the test fingerprint is compared against the fingerprints within the map. The results of comparing the test set fingerprints with the map are shown in Fig. 5. The experiments show promising results, as each test fingerprint's best match was the nearest map fingerprint node or an adjacent fingerprint node. We would like to see a test fingerprint situated halfway between two map fingerprints match equally well to both nearby nodes, but this is not always the case. Even though the map and test nodes look close in the figure, even a small spatial difference can cause differences in the features extracted or objects being detected, due to the relatively small area that each image covers. Another promising result of SPLINTR is shown in Fig. 6. The top image is the test image, and the middle and bottom images are the closest matches from the map based on the fingerprint similarity measures. There appears to be relatively few useful features

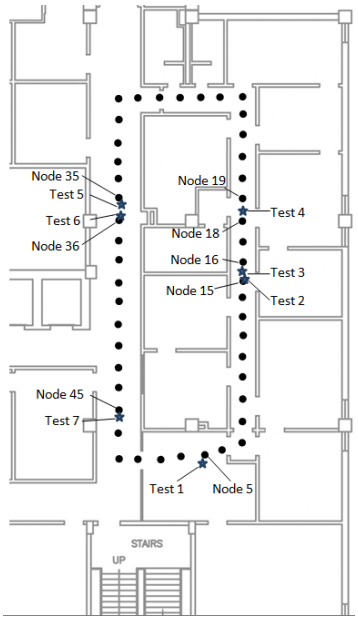


Fig. 4: The map consisted of 46 locations from the first floor of the Boyd Graduate Studies Research Center, indicated as black circles. The test locations are indicated using stars. The test nodes and their nearest map nodes are labeled.

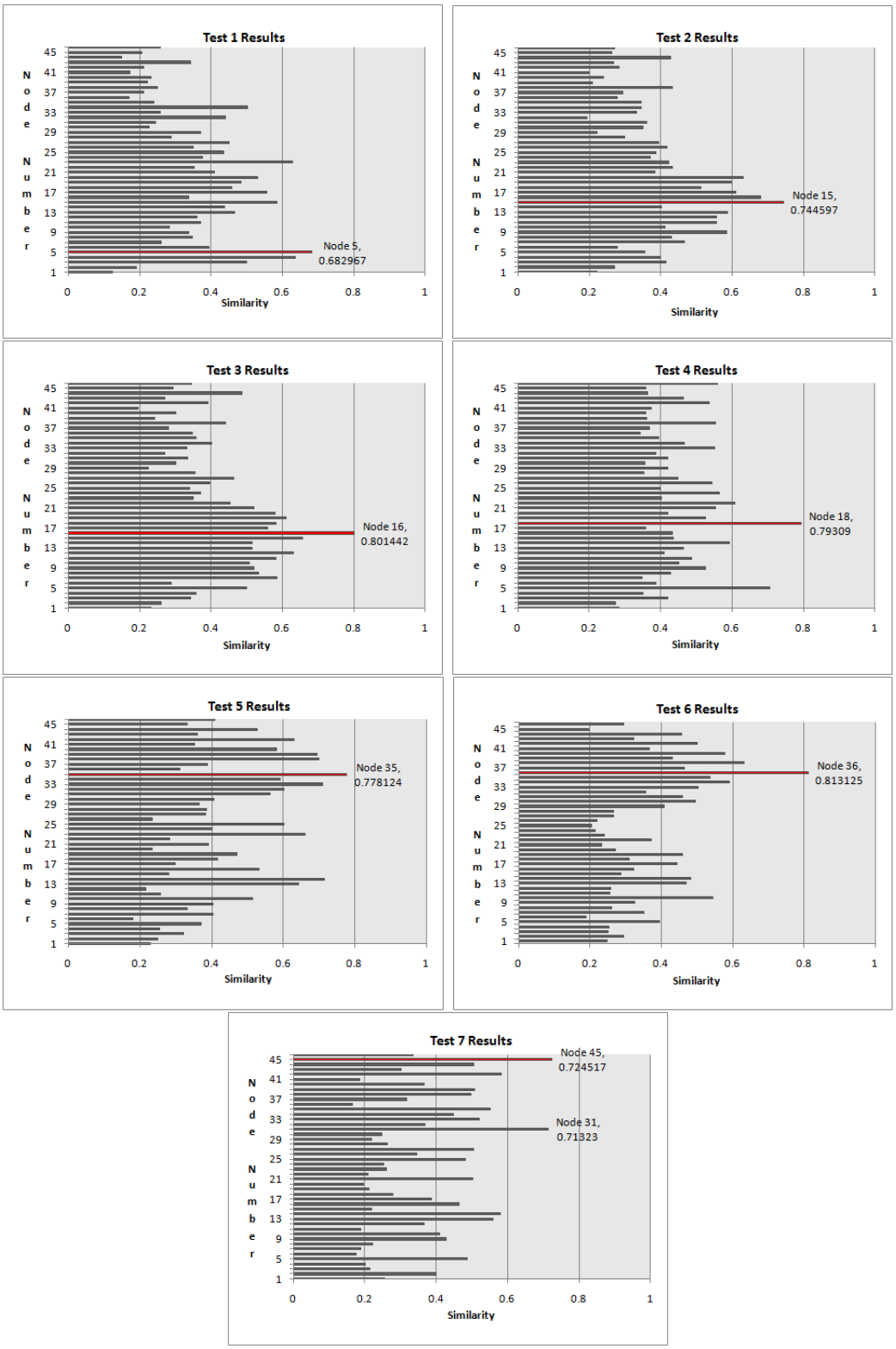


Fig. 5: Result graphs using the test fingerprints and map fingerprints shown in Fig. 4. The closest fingerprint match from the map and the similarity measure is indicated. Test 7 shows the closest match as well as the second closest match.

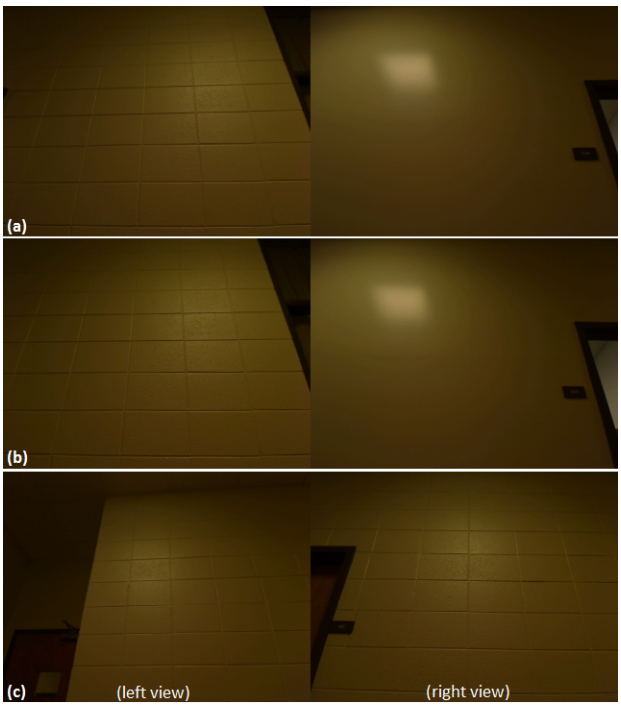


Fig. 6: (a) The query image, test 7. (b) Test 7's closest fingerprint match image from the map (Node 45) and (c) its second closest image match (Node 31). The left and right views from the robot's perspective are indicated.

with which to localize in the test image, as walls take up large areas of the image. As shown in the Test 7 Results graph in Fig. 5, however, the test 7 node matches the correct closest node in the map, and the next closest match as shown in Fig. 6 is a relatively close looking match, as well.

6. Discussion and Future Work

SPLINTR is our proposed system for a place recognition and topologically mapping system, forming visual fingerprint signatures of locations to be used on a mobile robot. We discussed the layout of SPLINTR as well as our implementation of the visual feature ridges of the system, and demonstrated SPLINTR's use in localization. The SPLINTR fingerprint method shows promising results when used in localization. Future work includes further tweaking of the individual fingerprint ridges and similarity measures. Continued steps in the realization of the SPLINTR fingerprint system include developing the system for topological mapping, as well as deploying the system on a mobile robot for autonomous topological map building and exploration.

References

[1] S. Se, D. Lowe, and J. Little, "Vision-based mobile robot localization and mapping using scale-invariant features," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2001, pp. 2051–2058.

[2] P. Neubert, N. Sunderhauf, and P. Protzel, "Fastslam using surf features: An efficient implementation and practical experiences," in *6th IFAC Symposium on Intelligent and Autonomous Vehicles (IAV)*, 2007.

[3] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[4] K. S. Saladin, *Anatomy and physiology: the unity of form and function*. McGraw-Hill, 2007, ch. 6, p. 196.

[5] P. Lamon, I. Nourbakhsh, B. Jensen, and R. Siegwart, "Deriving and matching image fingerprint sequences for mobile robot localization," in *IEEE International Conference on Robotics and Automation*, vol. 2, 2001, pp. 1609–1614.

[6] A. Tapus and R. Siegwart, "Incremental robot mapping with fingerprints of places," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Aug. 2005, pp. 2429–2434.

[7] M. Cummins and P. Newman, "Fab-map: Probabilistic localization and mapping in the space of appearance," *International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.

[8] J. Martinez-Gomez, A. Jimenez-Picazo, J. A. Gomez, and I. Garcia-Varea, "Combining invariant features and localization techniques for visual place classification: successful experiences in the robotvision@imageclef competition," *Journal of Physical Agents*, vol. 5, no. 1, pp. 45–54, January 2011.

[9] A. Pronobis and B. Caputo, "The robot vision task," in *ImageCLEF*. Springer Berlin / Heidelberg, 2010, vol. 32, pp. 185–198.

[10] F. Fraundorfer, C. Wu, and M. Pollefreys, "Combining monocular and stereo cues for mobile robot localization using visual words," in *International Conference on Pattern Recognition*, 2010, pp. 3927 – 3930.

[11] J. Hays and A. Efros, "Im2gps: estimating geographic information from a single image," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2008, pp. 1–8.

[12] A. Oliva and A. Torralba, "Modeling the shape of the scene: a holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–174, 2001.

[13] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *International Conference on Computer Vision*, vol. 2, July 2001, pp. 416–423.

[14] L. Xing and A. Pronobis, "Multi-cue discriminative place recognition," in *Multilingual Information Access Evaluation II. Multimedia Experiments*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2010, vol. 6242, pp. 315–323.

[15] R. Castle, D. J. Gawley, G. Klein, and D. W. Murray, "Towards simultaneous recognition, localization and mapping for hand-held and wearable cameras," in *International Conference on Robotics and Automation*, April 2007, pp. 4102–4107.

[16] P. Blaer and P. Allen, "Topological mobile robot localization using fast vision techniques," in *IEEE International Conference on Robotics and Automation*, May 2002, pp. 1031–1036.

[17] R. Hess, *Open-source SIFT implementation*. [Online]. Available: http://web.engr.oregonstate.edu/ hess

[18] N. Seo, *MergeVec*, 2007. [Online]. Available: http://note.sonots.com/SciSoftware/haartraining/mergevec.cpp.html

[19] J. Tarver, "Computer vision capabilities for a semi-autonomous wheelchair," Master's thesis, University of Georgia, 2008.

[20] J. Samarabandu and X. Liu, "An edge-based text region extraction algorithm for indoor mobile robot navigation," *International Journal of Signal Processing*, vol. 2, pp. 273–279, 2007.

[21] Algorithmist.com, *Longest Increasing Subsequence*.

[22] S. S. Skiena, *The Algorithm Design Manual*. Springer, 2008, ch. 8, pp. 289–291.

[23] Cristóbal Liñán, *cvBlob library for OpenCV*. [Online]. Available: http://code.google.com/p/cvblob/

[24] *Tesseract OCR 2.04*. [Online]. Available: http://tesseract-ocr.googlecode.com

[25] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Doklady Akademii Nauk SSSR*, vol. 163, no. 4, pp. 845–848, 1965, (In Russian. English version - Soviet Physics Doklady 10(8): 707-710).

Applying Dynamic Conditions to an Auction Behavior-Based Robotic Architecture

Bradford A. Towle Jr., and Monica Nicolescu
University of Nevada, Reno, Nevada, United States of America
towle@cse.unr.edu, monica@cse.unr.edu

Keywords: Behavior-Based, AI, Distributed, Auction, HRI

ABSTRACT

Robotic systems in the real world will work in dynamic environments and have to choose between multiple, and occasionally conflicting goals. In order to facilitate these two requirements an Auction Behavior-Based Robotic Architecture (ABBRA) was developed that allowed different behaviors to compete by bidding for control of a robot. Each behavior would bid with an activation level calculated from metrics based on the environment. This paper introduces two new features to the Auction Behavior-Based Robotic Architecture, which are the dynamic addition of a goal at run time and the ability to set time constraints on more than one behavior. Dynamic situations may require these new features in order to ensure a strict critical timing and adaptability to the environment. Coupling these two features together with the existing ABBRA system increased the goal selection performance across six test scenarios.

1. Introduction

Robots in a real-world environment can face several difficult problems. The real world is a dynamic environment that may impose time constraints, allow asynchronous input from humans and have situations where a robot may encounter multiple conflicting goals. Because of these issues, we developed an auction behavior-based robotic architecture (ABBRA)[1], which currently can adapt to most of these situations. Not only will ABBRA adapt to dynamic situations but it also attempts to choose the best goal in order to minimize execution time. To accomplish this, each behavior (module) must compete, or auction, with one another for control of the robot. Thus, this system uses a competitive winner-take-all action selection mechanism.

In addition to arbitrating between goals ABBRA can handle multiple conflicting goals requesting control simultaneously. Only a small number of robotic architectures can handle multiple conflicting goals without a static priority configuration such as the use of inhibition signals between behaviors where one behavior will deactivate another [2, 3].

Despite the above capabilities of ABBRA certain dynamic situations required the addition of two new dynamic features to the previously proposed auction behavior based robotic architecture [1]. The original architecture allows individual behaviors to challenge each other for control by bidding with an activation level. The activation level represents the importance of a certain goal. Metrics from the outside world derive the activation level, thus, the architecture can make the most opportunistic decision based on the most recent known values from the environment [1].

The two new features are as follows: i) the ability to allow temporal components into more than one behavior and ii) the ability to request multiple goals during runtime. Allowing the architecture to consider temporal constraints for more than just one behavior incorporates scheduling into the robotics decision making. The robot can now judge behaviors based on how much time they have left and determine whether they are critical or not. This also gives the robot the ability to postpone a lesser time-critical task in lieu of more opportunistic tasks. The second component, dynamic addition of a task during runtime, makes the architecture more robust against unknown changes and allows for a non-static architecture. Most behavior-based architectures require a recompilation or at least reset after adding a new module due to the usually static interconnectivity of the behaviors. However, the ABBRA system will simply add the new

module into the system, and start running it on the next program cycle.

The paper is structured as follows: Section II provides related work. Section III gives a more in depth look at the robotic architecture and the new capabilities added to it. Section IV presents the results taken from six different tests. Section V evaluates the findings. Section VI provides a brief look into the future work of this project followed by the conclusion.

2. Related Work

There are several major action selection mechanisms for robotic decision-making. Deliberative architectures plan and create a ideal solution for the known world [4, 5]. Problems arise when these systems encounter dynamic environments, because the robot must re-plan whenever it detects the situation has changed. As an alternative, voting allows each behavior choose the best action for itself. This works well when behavior outputs are similar enough to estimate which action is closest to each behavior's desired goal. To this end, the architecture must also select which actions will be voted on in the first place [6]. Arbitration is another option in which a robot has multiple behaviors, but must choose only one to execute. However, when multiple goals that conflict [2] arise, the architecture must prioritize [7] or select which behavior is most applicable [8].

The ABBRA project uses a winner-take-all method. In this approach, the robot must choose between a set of possible behaviors and choose the one it wishes to perform. The problem arise when handling conflicting goals and allowing the architecture to change goals when it is opportunistic for the robot. Activation Networks [2] solve conflicting goal by allowing behaviors to promote other behaviors by injecting "activation energy". The behavior that has the most activation energy will win control. Although similar, ABBRA does not use inter-behavior communication to promote activation for a certain behavior. Instead, ABBRA uses the environment to determine which behavior is most efficient to run. This follows the standard behavior based paradigm where data from the environment provides state information [9]. Generally, behavior based paradigms will use this information to prioritize goals [7] or use inhibition signals to prohibit conflicting goals [3]. ABBRA extends this concept, instead of simply prioritizing goals, it will allow them to compete and dynamically change their priority.

Market-based approaches are widely used in multi-agent robotic systems. Since the seminal paper [10] the number of market based robotic papers has increased dramatically increased [11-20]. However, these papers focus on multi-agent (multi-robots) and solve a different problem than ABBRA. Here are some key differences: 1) The first difference is that multi-agents systems involve robots competing for task where as ABBRA deals with behaviors competing for control over actuators or other robotic resources; 2) Multiple agents bid for tasks whenever they becomes available where as in ABBRA the bidding occurs continually; 3) Multiple agent systems must monitor the robot who won the tasks to ensure that it is performing well [13] - in ABBRA if a task does not perform well another task will out-bid it on the next cycle; 4) Multi-agent systems must worry about external conflicts between robots, where ABBRA resolves conflicting goals by allowing environmental and temporal metrics to influence which behavior has won [21]; 5) Conversely, a lot of research has been done to allow individual agents to cooperate with each other – in ABBRA behaviors that are simultaneously running will automatically be capable of helping other behaviors [14]. Because of these differences, the problem ABBRA solves and the domain of multi-agent systems are substantially different.

3. The Architecture

ABBRA allows multiple behaviors to challenge each other for control of an actuator in order to accomplish their specific objectives. Each behavior calculates an activation level that describes its priority to run. The activation level is calculated by multiple metrics form the environment. These metrics represent the "closeness" of the desired goal state to the current state of the robot. This ensures that the activation level describes the most up-to-date environment as possible [9]. Each metric is divided by the maximum value of that particular metric's type. All active metrics in a behavior are then summed and averaged resulting in a normalized activation level that can be compared to all other behaviors. Equation 1 (next page) details this calculation.

Subtracting from one inside the summation allows the user to specify that a lower metric value is more desirable. Without it a higher metric value would yield a higher activation level. After calculating the activation level, each behavior will bid for control of a mutex, which grants access to various robot actuators or any other limited resources. The highest bid wins control for that round. This causes an emergent behavior where the robot will try to handle the

easiest/most important tasks first [1] yet it is still capable of changing behaviors if necessary.

$$\text{Activation Level} = \frac{\sum_{All\ metric\ types} 1 - \frac{BehaviorsMetricValue}{Max\ value\ for\ Metric\ type}}{Number\ of\ used\ Metrics}$$

Equation 1: Normal calculation for activation levels.
BehaviorsMetricValue - Value of a particular metric for a particular behavior wishing to bid for an actuator.
MaxValueforMetricType - The maximum value of the particular metric for all behaviors challenging.
NumberofUsedMetrics - The number of metrics used in the particular challenging behaviors.

It is desirable that the robot using ABBRA could handle multiple goals with varying degrees of time constraints. For example, behavior "A" must end in five minutes, but behavior "B" must end within 35 seconds, the robot should choose behavior "B" with all other metrics being equal. However, if two goals have large time constraints while only requiring a small amount of time, there should be very little contribution from the time constraints to the activation level. Thus, equation 1 does not work. Consider a scenario where two goals both take five minutes to complete: one has a time constraint of 30 minutes and the other has a time constraint of an hour. The time constraint is almost pointless in this scenario. However, if the time constraint is closer to zero then the it should be important even if the difference between another goal is only a couple of minutes. In order to handle this situation the architecture uses a different fitness function for the temporal metric. The percentage of time remaining applied to Equation 2 (next column) gives the appropriate non-linear response to the activation level. Originally, the time-to-finish was to be used. However, many behaviors' completion time are difficult to estimate. Therefore, we used the time-to-start metric where the time before a behavior must execute can be estimated and used in the activation level calculation.

$$ContributionFT = e^{-10*(\mathrm{PRS})}$$

Equation 2: Calculating the contribution for time remaining
PRS - The percentage of time that remains before the task MUST start.
ContributionFT = This the value that will be added to the summation in equation 1 for time instead of equation 1.

Another newly added component of the ABBRA is the capability for the architecture to add a new goal into the system after the system started running. This allows the architecture to add new goals for situations it did not know existed. This process was straightforward for the ABBRA system. The user, or another program, adds a new goal, which in turn adds its own behaviors into the architecture. The next bidding cycle calls these behaviors as if they have been there since the robot had started running, and then the new behaviors begins to compete for control. Currently no inter-connecting inhibition signals are required because ABBRA does not use inhibition signals.

4. Experimental Results

Six scenarios were designed to test the new features. There are six goals in each test, each containing five locations that the robot must navigate using a "Go-To-Point" behavior. The sixth goal uses "Center-on-Green object", which centers the robot at a green object when one is found. The first tests provided a simple control run containing no time constraints, and no goals were added at run time. The second scenario tested the possibility of having two known goals (locations five and six) with a strict time constraint. Scenario three provided the same analysis as scenario two except that this time the location of the goal was unknown because it gave "center on green object" a stricter time constraint. Tests four and five dynamically added a new goal during runtime. This demonstrated what would happen if the architecture added a goal with an urgent time constraint. Test four added "Center-On-Green-Object" midway through the scenario execution, likewise scenario five added "Go-To-Point" on goal five during the execution. The sixth scenario added the same goal as scenario five however, no time constraints were placed on the new goal. These tests were run on the player/stage package simulating a Pioneer 3 robot.

4.1 Experiment 1: No Constraints

The first test demonstrates a typical run where the robot must travel to five goal locations and center itself and its camera on a green object, whose location is unknown. Figure 1 details the path the robot took to reach the goal.

This first run contained no time constraints, thus no goal had a smaller scheduled time for completion than any other goal. Therefore, the robot takes a "shortest distance first" approach and centers itself on the first green object it can find. Figure 1 (bottom) details the order in which the processes ran. Without any interruptions or time constraints, the robot

simply executed the closest goal first interrupting Goal 1 only to center on the green object that it detected.

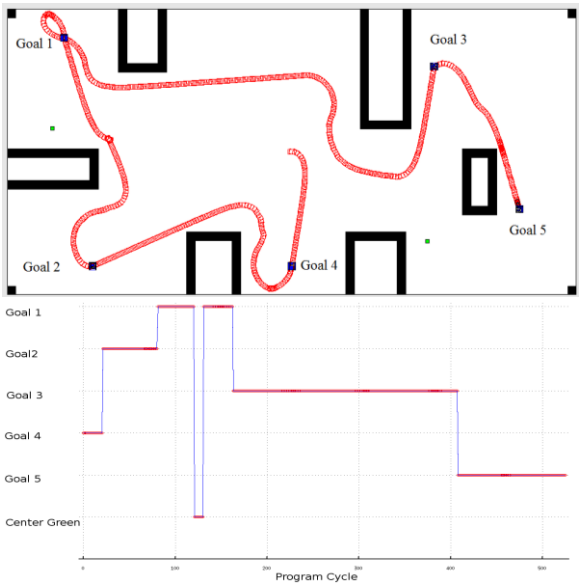


Figure 1: Path taken by on the first run (Top), Order of goals taken (Bottom)

4.2 Experiment 2A : Time Critical Constraint for Center on Green Object

The second set of tests focused on goals with time constraints. The first time-critical test configures Goal 1 and Goal 5 with a very urgent time constraint. The time constraints are set so it reaches zero immediately. In real life, this situation would be unlikely but it would be equivalent to having two emergencies occurring at once.

The following images show the order in which the processes of Test 2A executed. The robot immediately moves to finish Goal 5 and then goes toward Goal 1. The robot did not visit Goal 3 although it came in close proximity to this goal. This demonstrates the importance the robot put on Goal 1.

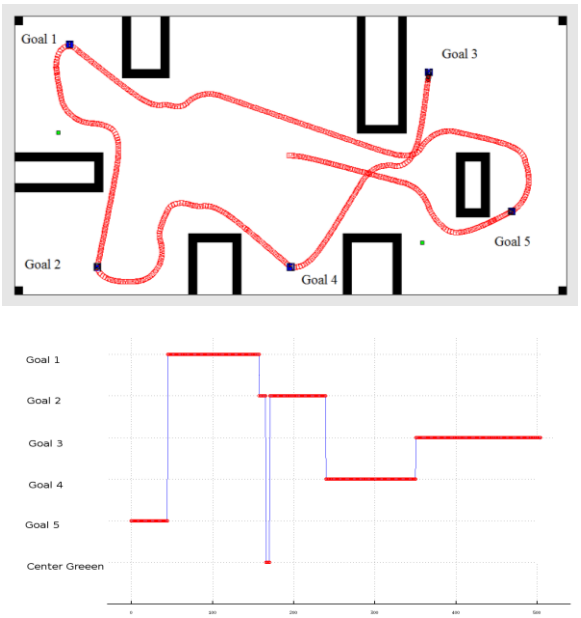


Figure 2: Path for Test 2A (Prev. Page), Order for Test 2A (Top)

4.3 Experiment 2B: Time Critical Constraint for Goal 5

Part B of the second test was setting an unknown goal, such as "Center-On-Green-Object", to have a time critical requirement. In the following figure the "Center-On-Green-Object" is running instead of "Go-to-Point", therefore, it uses a slightly different obstacle-avoid procedure designed to wander a maze. The robot did encounter Goal 3 - However, this was purely coincidental. Notice the robot avoids Goal 5 until finally discovering the green object in the right hand corner. Figure 3 (next page top) details the path of the robot and Figure 3(next page bottom) shows the order of goal fulfillment.

Figure 3: Path for Test 2B (Top), Process order for Test 2B (Bottom)

4.4 Experiment 3A: Dynamically Adding Center on Green Goal with Time Constraint

The third test introduces a new goal with an urgent time constraint during run-time. A common problem in robotics is that once an architecture has started to run it cannot dynamically add modules to it. The ABBRA system handles it as a new competitive module. The following figure details the path taken by the robot: As it is rounding the corner for Goal 1 the "center on green object" behavior is added, thus the behavior is spawned. The process order of the third test shows where the "Center-On-Green-Object" started and when it interrupted the first goal. Note on Figure 4 (Bottom) the blue "X" marks the step in which "Center-On-Green-Object" dynamically added itself to the architecture.

4.5 Experiment 3B: Dynamically Adding Goal 5 with Time Constraint

The second portion the third test repeated the same test, but this time it would dynamically add Goal 5. The results for this are more dramatic. As the robot goes around the corner to reach goal one, Goal 5 with a short time constraint is created. This forces the robot to abandon the current goal and travel across the map to Goal 5 (Figure 5).

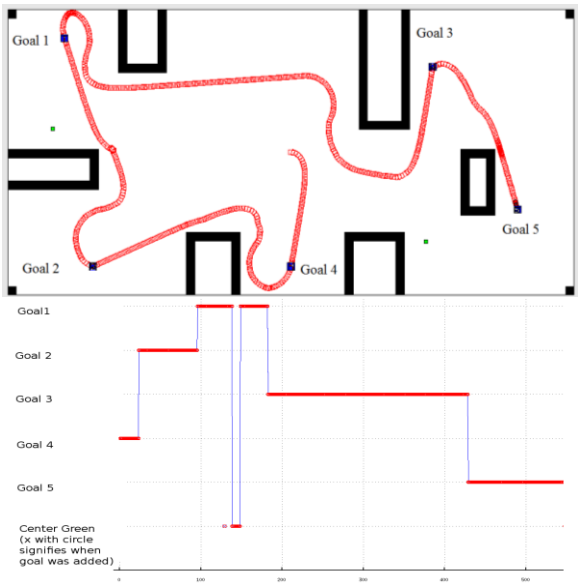


Figure 4: Path for Test 3A (Top), Order of goals for 3A (Bottom)

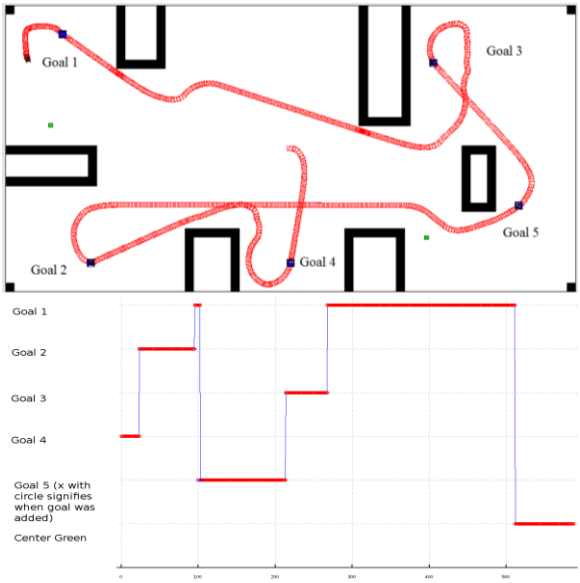


Figure 5: The path taken by the robot for Test 3B (Top), Order of goals for Test 3B (Bottom)

The blue "x" close to the 100 mark on Figure 5 (Bottom), mark indicates the step in which Goal 5 was added to the system. Since the behavior associated with goal five has an urgent time constant, it immediately takes over and begins running.

4.6 Experiment 4: Dynamically Adding a Goal without Time Constraint

In this test, Goal 5's addition occurs at the same time as in the previous test; however, Goal 5 has no time critical component. This test demonstrated that the addition of a goal with relative equal temporal priority would have very little effect on the system. As the results show, Goal 5 has no impact when added with no time constraints (Figure 6). Figure 6 (top) details Goal 5 being added and no real change occurring to ABBRA's order of goal fulfillment.

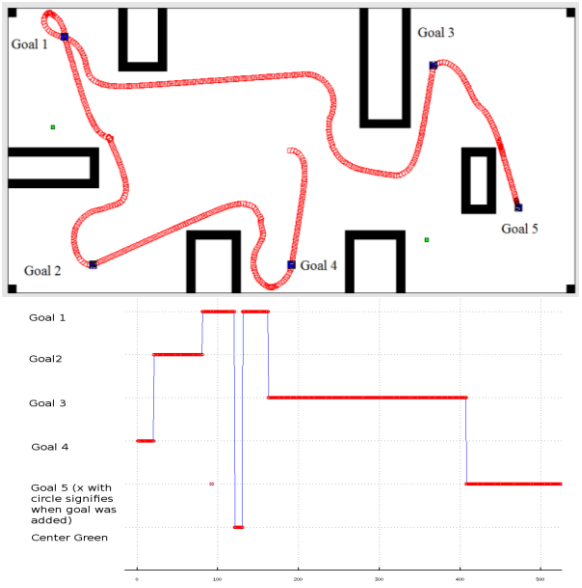


Figure 6: Path taken for Test 4 (Top), process order for Test 4 (Bottom)

5. Discussion

The data shown above supports the following two claims: first, allowing the robot to consider temporal constraints gives the robot a much more adaptive ability. For example, the robot may change the time priority of a goal in case an emergency arises. The time constraint could also allow the robot to have safety thresholds. For instance, if a robotic arm dealing with chemicals required the addition of extra chemicals with accurate timing, the architecture could consider this. Allowing multiple time constraints gives the user the ability to specify that a goal can execute whenever there is no other time critical goals challenging. The user could also ask the robot to do something when it was convenient - if the robot had pre-existing goals that contained a time critical component, it would finish those goals first.

Second, the ABBRA also allows the robot to handle additional goals given to it after startup. This is necessary in order to ensure that the robot can adapt to new challenges, requests and obstacles. One of the most common use cases for adding a goal dynamically is allowing the user the ability to add a new task requirement for a robot. This could be through any medium as long as the goal is registered and introduced into the system. The robot may also detect an obstacle that was unexpected, and overcoming this obstacle may require dynamically adding a new task.

6. Future Work

This paper demonstrates the importance of the ABBRA's adaptability. The next step of this work incorporates this system into the real world by testing it on a physical robot. Future work is also needed regarding human robot interaction (HRI). If humans interact with the robot, the architecture must choose between performing a task and interacting with the human. A way of dealing with this consists of using teamwork between the robot and human [25-30]. However, the user must still interact with the robot in terms of behaviors and goals; this requires the architecture to fuse low level behaviors into a level of abstraction users want to interact in [31]. Incorporating SLAM (Simultaneous Localization and Mapping) increases the accuracy of the architecture substantially. If the robot can identify real distance to goals versus straight-line distance, the architecture will have a more accurate understanding of its current state in the environment.

7. Conclusion

In conclusion, the ABBRA gives the robot the ability to adapt to dynamic environments. By allowing temporal constraints in the behavior's activation level calculation, the robot can prioritize a task based on temporal requirements. This paper demonstrates the ability for the robot to introduce a new goal into the architecture that was not initially running at the beginning of runtime. The robot can use these new features coupled with the older features provided by the ABBRA to adapt to a wide variety of dynamic environments as demonstrated by the scenarios presented.

8. References

[1] B. A. Towle, and M. Nicolescu, "Fusing Multiple Sensors through Behaviors with the Distributed Architecture," in 2010 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, Salt Lake, Utah, 2010, pp. 115-120.

[2] P. Maes, "How to do the right thing," *Connection Science*, vol. 1, no. 3, pp. 291-323, 1989.

[3] M. Proetzsch, T. Luksch, and K. Berns, "Development of complex robotic systems using the behavior-based control architecture iB2C," *Robotics and Autonomous Systems*, vol. 58, no. 1, pp. 46-67, Jan, 2010.

[4] R. Volpe, I. Nesnas, T. Estlin *et al.*, "The CLARAty architecture for robotic autonomy," in Aerospace Conference, 2002, pp. 1.

[5] T. Estlin, R. Volpe, I. Nesnas *et al.*, "Decision-making in a robotic architecture for autonomy."

[6] J. Rosenblatt, and C. Thorpe, "Combining multiple goals in a behavior-based architecture." pp. 136-141.

[7] R. A. Brooks, "A ROBUST LAYERED CONTROL-SYSTEM FOR A MOBILE ROBOT," *Ieee Journal of Robotics and Automation*, vol. 2, no. 1, pp. 14-23, 1986.

[8] J. Koseck , and R. Bajcsy, "Discrete event systems for autonomous mobile agents," *Robotics and Autonomous Systems*, vol. 12, no. 3-4, pp. 187-198, 1994.

[9] R. Brooks, "Elephants don't play chess," *Robotics and autonomous systems*, vol. 6, no. 1-2, pp. 3-15, 1990.

[10] R. Davis, and R. G. Smith, "Negotiation as a metaphor for distributed problem solving," *Artificial intelligence*, vol. 20, no. 1, pp. 63-109, 1983.

[11] F. Brandt, W. Brauer, and G. Weiss, "Task assignment in multiagent systems based on vickrey-type auctioning and leveled commitment contracting," *Cooperative Information Agents IV-The Future of Information Agents in Cyberspace*, pp. 11-44, 2004.

[12] P. Faratin, C. Sierra, and N. R. Jennings, "Negotiation decision functions for autonomous agents," *Robotics and Autonomous Systems*, vol. 24, no. 3-4, pp. 159-182, 1998.

[13] B. P. Gerkey, and M. J. Mataric, "Sold!: Auction methods for multirobot coordination," *Robotics and Automation, IEEE Transactions on*, vol. 18, no. 5, pp. 758-768, 2002.

[14] B. Jennings, and Å. Arvidsson, "Co-operating market/ant based multi-agent systems for Intelligent Network load Control," *Intelligent Agents for Telecommunication Applications*, pp. 71-71, 1999.

[15] H. Jung, M. Tambe, and S. Kulkarni, "Argumentation as distributed constraint satisfaction: Applications and results." pp. 324-331.

[16] R. Krovi, A. C. Graesser, and W. E. Pracht, "Agent behaviors in virtual negotiation environments," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 29, no. 1, pp. 15-25, 1999.

[17] M. J. Matari , G. S. Sukhatme, and E. H. Østergaard, "Multi-robot task allocation in uncertain environments," *Autonomous Robots*, vol. 14, no. 2, pp. 255-263, 2003.

[18] R. G. Smith, "The contract net protocol: High-level communication and control in a distributed problem solver," *Computers, IEEE Transactions on*, vol. 100, no. 12, pp. 1104-1113, 1980.

[19] K. Sycara, and D. Zeng, "Coordination of multiple intelligent software agents," *International Journal of Cooperative Information Systems*, vol. 5, no. 2, pp. 181-212, 1996.

[20] M. P. Wellman, and P. R. Wurman, "Market-aware agents for a multiagent world," *Robotics and Autonomous Systems*, vol. 24, no. 3-4, pp. 115-125, 1998.

[21] M. B. Dias, and A. Stentz, *Traderbots: A market-based approach for resource, role, and task allocation in multirobot coordination*: Citeseer, 2003.

[22] D. Ferguson, Y. Yemini, and C. Nikolaou, "Microeconomic algorithms for load balancing in distributed computer systems." pp. 491-499.

[23] L. M. Ni, C. W. Xu, and T. B. Gendreau, "A distributed drafting algorithm for load balancing," *Software Engineering, IEEE Transactions on*, no. 10, pp. 1153-1161, 1985.

[24] I. Ahmad, and A. Ghafoor, "Semi-distributed load balancing for massively parallel multicomputer systems," *IEEE Transactions on Software Engineering*, pp. 987-1004, 1991.

[25] T. Fong, N. Cabrol, C. Thorpe *et al.*, "A personal user interface for collaborative human-robot exploration."

[26] T. Fong, C. Thorpe, and C. Baur, *Collaborative control: a robot-centric model for vehicle teleoperation*: Carnegie Mellon University, The Robotics Institute, 2001.

[27] T. Fong, C. Thorpe, and C. Baur, "Multi-robot remote driving with collaborative control," *Ieee Transactions on Industrial Electronics*, vol. 50, no. 4, pp. 699-704, 2003.

[28] T. Fong, C. Thorpe, and C. Baur, "Robot, asker of questions," *Robotics and Autonomous Systems*, vol. 42, no. 3-4, pp. 235-243, 2003.

[29] G. Dorais, R. Bonasso, D. Kortenkamp *et al.*, "Adjustable autonomy for human-centered autonomous systems."

[30] A. Fereidunian, M. Lehtonen, H. Lesani *et al.*, "Adaptive autonomy: smart cooperative cybernetic systems for more humane automation solutions." pp. 202-207.

[31] M. Nicolescu, O. Jenkins, and A. Stanhope, "Fusing robot behaviors for human-level tasks." pp. 76-81.

A COG Analysis Model of System-of-Systems (SoS) Based on Multi-Entity Bayesian Networks (MEBN)

Zhou Yun, Zhu Cheng, Lei Ting, Zhang Weiming, Liu Zhong
Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha, Hunan, China

Abstract - *The centre of gravity (COG) is the source of power and the most important system of system-of-systems (SoS). A detailed study of COG analysis of SoS has been carried out by using multi-entity Bayesian networks (MEBN) method. The MEBN can be converted into different domain specific Bayesian networks according to different inputs, which is very useful to tackle with the uncertain behaviors of SoS. Then the COG can be measured with Bayesian network inference. In the experiment, we build a COG analysis model of an alert SoS, which considers the repair and backup behaviors of radar system under attack. Degradations of COG are analyzed in the case study, which reflect the effectiveness of different inputs (attack strategies). The results show that degradations are different according to different attack strategies, which support generations of attack plans. This analysis model can help the commander to make the more effective decisions in future warfare.*

Keywords: System-of-systems (SoS); Multi-entity Bayesian networks (MEBN); COG analysis model; Decision support.

1 Introduction

Most huge scale integrations (such as organizations, nature environment, biologic systems and etc.) can be recognized as system-of-systems (SoS). Eisner [1] firstly proposed SoS concept during his research on system integration. From then on, it has become an important topic of research in many domains, which has at least 40 definitions. Here we define the SoS is the integration of independent operational systems, where each system could achieve its goal and make contributions to SoS, such as operation SoS. Centre of gravity (COG) is the source of power, which represents the strength of operation SoS. Nowadays, the COG analysis model that considers behaviors of SoS under attack is a new problem that need to be tackled. Researches on invulnerability of complex networks, complex adaptive system (CAS), and influence nets afford the foundation to solve this problem.

The attack tolerance of complex networks is begun with the Albert's research in 2000 [2]. It is said that many complex networks are invulnerable to random attack. But they are very vulnerable to the attack on most connected nodes. These researches considered how to design the networks to keep the invulnerability when some nodes or edges are removed. However, this method is based on structure design that is not appropriate in SoS, as for diversities of systems.

CAS theory applies multi-agent technology to explore the combat behavior between SoSs. Carley [3] proposed the dynamic network analysis concept by using multi-agent technology. She applied this technology to evaluate the COG of a terrorist network, and analyzed the behaviors of the network under situations of isolating individuals at random, isolating the most connected individual, and isolating the highest cognitive load one. However, this method is only useful to the system that could make its own decisions in the SoS. But in operation SoS, only part of systems could make decisions, so this method is also limited here.

As there are causalities between systems in SoS, we could analyze the probability distributions of them, which include the COG. Jouni Pousi [4] has analyzed an operation SoS in strategic, operational, and tactical level with technologies of influence diagrams (ID) and multi-criteria decision analysis (MCDA). Then he generated the effects-based operations (EBO), which seek the most degradation of COG. Lucia Falzon [5] has also applied this method to build the Bayesian networks to analyze the COG, but the conditional probabilities are hard to specify. Ugur Kuter [6] improved the Causal Analysis Tools (CAT) of US Air Force Research Laboratory based on Bayesian Networks (BNs), and afforded the upper and lower bounds of successful probabilities of attack strategies. However, all these methods didn't take the behaviors of SoS into consideration. Because typical Bayesian network cannot depict these information.

Laskey [7] proposed the Multi-Entity Bayesian Networks (MEBN), which could specify these uncertain information with first-order language (FOL), and construct the modularized domain model that called MEBN fragments (MFrags) based on inputs. This method has been well used in the military domain such as situation assessment [8], tactical intention recognition [9], and etc. However, it has not been used in the SoS application. This paper work out the COG analysis model of SoS, firstly

build the behavior models of SoS with MFrags, secondly propose the combination algorithm of MFrags which could generate correlative Bayesian network based on different inputs, thirdly operate inference to analyze the COG, and finally carry out a case study to test this method.

2 COG analysis model of SoS

2.1 Multi-Entity Bayesian Networks (MEBN) and MFrags

The world in MEBN is composed of interrelated entities which contain attributes. The entities and their attributes are depicted and instantiated by FOL in MEBN. We can use different random variables to instantiate the entities for different background. For example, *Radar(r)* represents the radar entity designated by the variable *r*. *Damage(d)* represents the damage on radar designated by the variable *d*. *Time(t)* represents the repair time designated by the variable t. To refer to the degree of repair in time slice *t*, we would fill in values for *r, d* and *t* to obtain an instance *RadarSelf-repair(r,d,t)* of the *RadarSelf-repair* random variable. So we could build an effective model for the repair behavior of radar under attack with this method. Then the expressive power of BNs is extended.

The basic element in MEBN world is entity. The entity is depicted by constant variable, and the attribute is represented by random variable. In MEBN, the MFrags depict the causal relations and condition distributions. It is the model of domain knowledge. In MFrags, the node represents the random variable, and the arc represents the dependent relation between nodes. The definition of MFrags [7] is listed below.

Definition 1: A MFrag is defined as a tuple F = (C,I,R,G,D). Finite set C represents the context value assignment terms. Finite set I represents the input random variable terms. Finite set R represents the resident random variable terms. G is the acyclic directed fragment graph in a MFrag. And set D contains the local distributions of each member in R. The sets C, I, and R are pairwise disjointed. The nodes in G are in one-to-one correspondence with the random variables in I $\cup$ R. Random variables in I correspond to root nodes in G.

MFrags have strong expressive power that can be categorized at other three types: Recursive MFrags, Logical Connective MFrags, and Quantifier MFrags [7]. In Recursive MFrags, there are directly or indirectly influences between instances of a random variable and other instances of the same random variable. Figure1 (a) shows the transformation between two instances of self repair radar. Logical Connective MFrags can directly represent the logical relationship of random variables. Figure1 (b) shows that the random variable $\wedge(r)$ and its ancestors $\Diamond(r1)$ and $\Diamond(r2)$. According to the connection, the value of $\wedge(r)$ is T if both $\Diamond(r1)$ and $\Diamond(r2)$ have value T, F if $\Diamond(r1)$ or $\Diamond(r2)$ has value F, and $\perp$ otherwise. Quantifier MFrags have quantifier random variables like $\forall(r,\phi r)$ or $\exists(r,\phi r)$. Figure1 (c) depicts an example shows that there is at least a distance between any radar and backup radar.

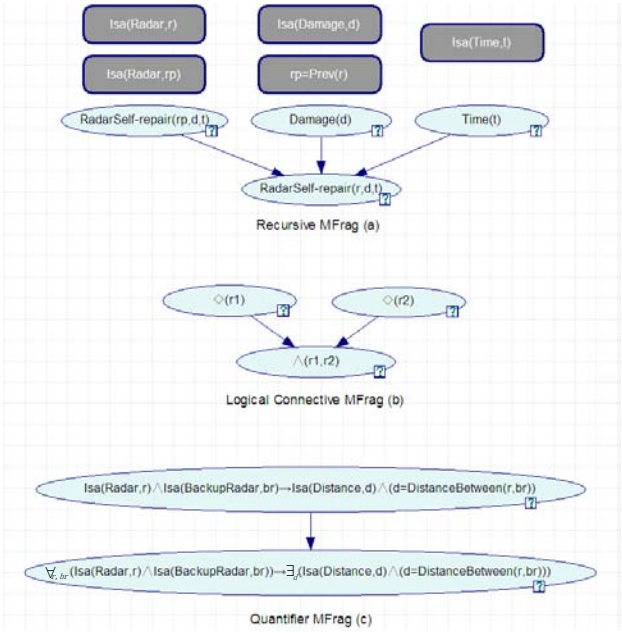


Fig. 1. Three Types of MFrags

2.2 Self-adaptive behaviors of SoS

If we want get more effective COG analysis, behaviors of SoS must be taken into consideration that can be represented in MFrags. In this section, we will outline the basic reaction behaviors of SoS under attack, such as repair behavior and backup behavior. These behaviors will be modeled in MFrags according to the method in section A.

Figure 2 is the MFrag that shows the repair behavior of radar under attack. There are seven nodes in this figure, 3 of them are context nodes, 3 are input random variable nodes, and one is resident random variable node. Parameter *d* and *t* represent the damage degree and repair time respectively.

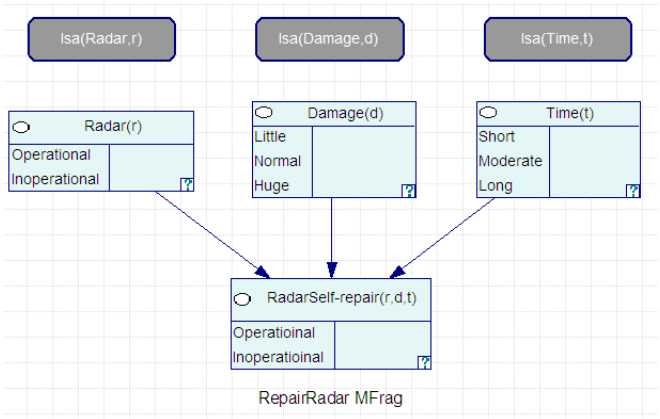


Fig. 2. MFrag of The Repair Behavior

The backup behavior is depicted in Figure 3. There are six nodes in this figure, 3 of them are context nodes, 2 are input random variable nodes, and one is resident random variable node. Parameter *d* represents the distance between the original radar and the backup radar. When all the behaviors of attacked radar are depicted by the MFrags, we can say that the domain-specific knowledge base is successfully built.

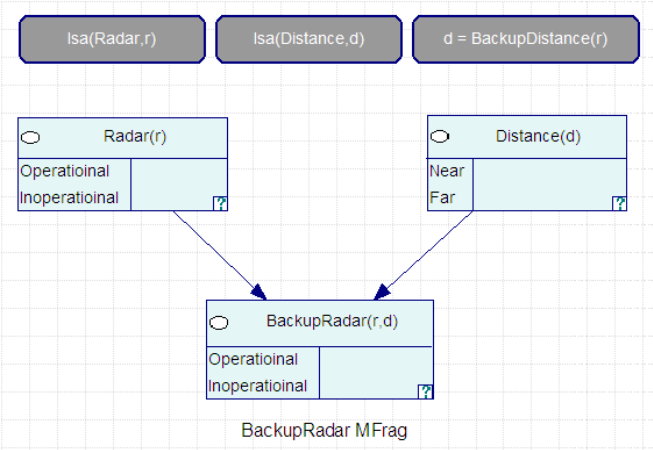


Fig. 3. MFrag of The Backup Behavior

2.3 BNs construction algorithm based on inputs

BNs inference supports the COG analysis, which is the final goal of MEBN. We can construct the domain-specific Bayesian network from the MFrags according to different inputs. That means we can get different Bayesian network under different attack strategies.

After we get the Bayesian network, the standard Bayesian network inference algorithms (such as message-passing algorithm, junction tree algorithm, stochastic sampling algorithm and etc.) can be used to infer. Mahoney and Laskey [10] proposed the simple bottom-up construction algorithm, which begins with a query and finding set of target random variable instances. Then it is followed with recursive processes that the ancestors of target random variable instances are added in the BNs. Here we outline another construction algorithm based on their algorithm and domain characteristic of this problem, which is showed in Figure 4.

Step 1 Initialization. Set the attack strategies set A, for each $a \in A$, $a = (r_i, r_j, ..., r_k)$, here the *i*, *j*, *k* are the identification of the objects which could be attacked. Set two finite integers: $n = 0$, which represents the size of set A; $m = 0$, which represents the number of element in *a*. The set BN is the final Bayesian network set that we need. Then the structure and local distribution construction can be described in step 2 with pseudo-code.

Step 2 Structure and Local Distribution Construction.

For (n=0; n<size(A); n++)
 For (m=0; m<element_number(a); m++)
  Set a_m = target_node(BN_{nm}) ;
  Find_ancestors(a_m);
   If(ancestors(a_m) $\notin$ findings set){Quit()};
   Set_distributions(a_m , ancestors(a_m));
 End For
 BN_n =
   Combine($ftarget, BN_{n1}, BN_{n2}, \ldots, BN_{nm}$);
End For

Step 3 Inference. Standard inference algorithm is used to compute conditional distributions for the final target node (COG) given the ancestor nodes in each BN_n .

Fig. 4. The Domain-specific Bayesian Network Construction Algorithm

In step 2, each attack strategy a has an order of attack targets. For each attack target, we can get its Bayesian network. So the final target node (the centre of gravity node of SoS) is specified in the combine function, which combines all target BNs into an integrated one for the COG. That means each attack strategy has its own integrated Bayesian network. Additionally, the same COG is destined in each integrated Bayesian network. So the status changes of COG can reflect the effectiveness of different inputs (attack strategies). Then we can get results of analysis.

3 The case study of COG analysis

3.1 Description and MEBN construction

Here is a case study that applied MEBN, which considers the repair and backup behaviors of nodes in the alert SoS. The strategic objective for the red country is to conduct air-to-ground operations against the alert SoS of blue country. It is assumed that the alert SoS has two types of systems: a command center, and some ground-based radar. What's more, the command center are seriously protected that the cost is too high to attack this target, so the red side only attack the ground-based radar. Here the number, defendable ability, self repair ability, backup, and detective ability is all uncertain. So we can use MEBN to build the domain-specific model. The detailed MFrags are listed in Figure 5.

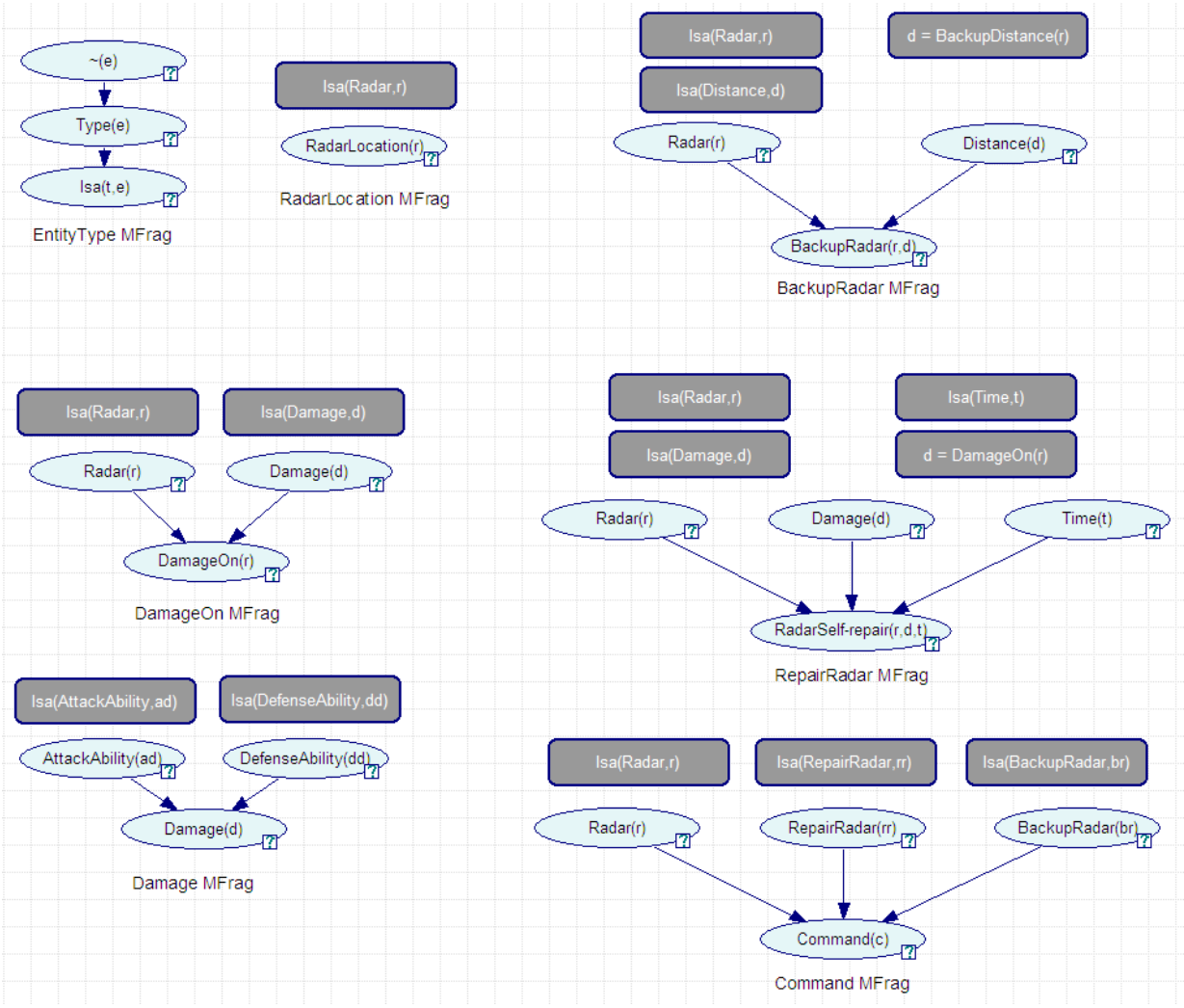


Fig. 5. MFrags for Alert SoS under Opponent Situation

In Figure 5, random variable ~(*e*) could be simply described as *e*. Then Isa(*t*,*e*) represents that *e* is an instance of *t*. As depicted in Command MFrag, here the ability of command center is affected by three types of radar: well operational radar, repaired radar, and backup radar. In RepairRadar MFrag, we find that when the radar experiences the attack, it will be repaired according to the damaged degree and repair time. What's more, the BackupRadar MFrag shows that when the backup behavior is triggered, the ability of backup radar is affected by the distance between original radar and backup one. Here we assume that if the ability of repaired radar reduces below 50%, then the backup radar will be operated.

3.2 Comparisons of two attack strategies

Now the intelligence agency of red country finds there are three radars (r1,r2,r3) controlled by the command center. The importance (IP), defendable ability (DA, the total is 10), self repair ability (SR, 50% means it can repair half of ability in one time slice), backup distance (BD) of each node is listed in Table 1.

Table 1 The Information of Radar in Blue Country

Number of Radar	*IP*	*DA*	*SR*	*BD*
r1	50 %	3	50 %	Near
r2	30 %	2	25 %	Near
r3	20 %	2	30 %	Far

In addition, the red country has two ballistic missile. The attack ability of the missile is 9. Here the strategic goal of red side is to decrease the ability of alert SoS with two operations (The time interval between two operations in this scenario is one time slice, and the maximal time interval is two slices). According to the permutation and combination theory, there are $3^2 = 9$ strategies. Let's focus on the first strategy that two ballistic missile successively attack the r1 radar. We could get the corresponding Bayesian network (Figure 6) based on the previous algorithm.

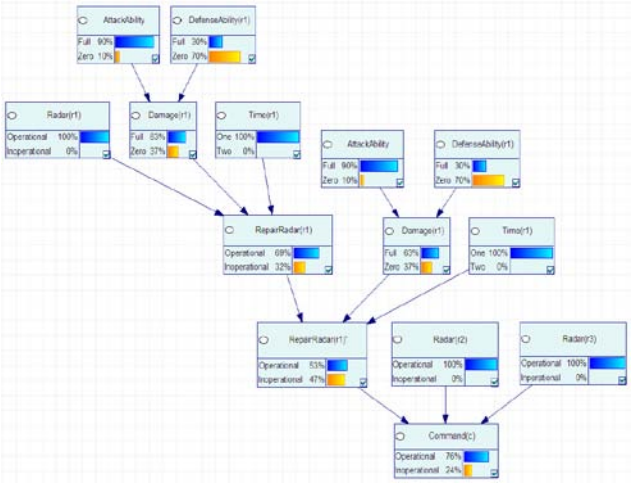


Fig. 6. Bayesian Network for The First Attack Strategy

The finding random variables are outlined here: the r1 radar is well operational before the attack, the r2 and r3 radar are both well operational after the attack. According to the standard Bayesian network inference algorithm, we find that the ability of commander center reduces to 76% (24% off) after a time slice behind the end of the second operation. As in [5], the centre of gravity (COG) of this alert SoS is the commander center, so the reduction coincide with the decrease of SoS's ability.

The second strategy is that one ballistic missile attacks the r1 radar, and another one attack the r2 radar. Here we find the r2 radar can only recover to 46% ability (less than 50%) in a time slice, when it is attacked. So this will trigger the backup mechanism of r2. Because the position of backup radar is near from the original r2 radar, we assume that the backup radar will get 80% ability. Then we could get the corresponding Bayesian network that is showed in Figure 7.

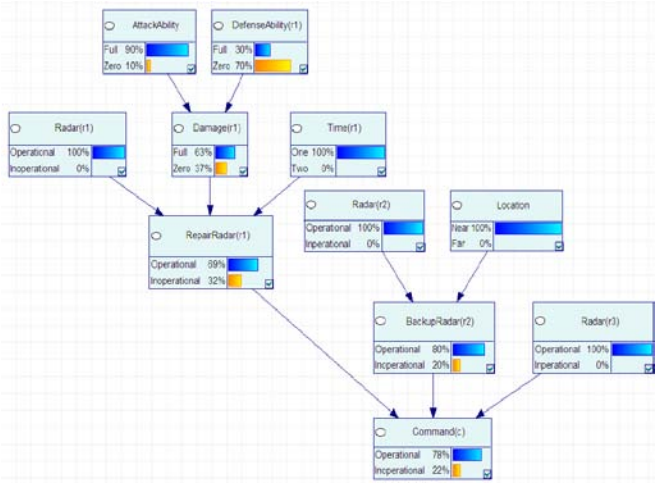


Fig. 7. Bayesian Network for The Second Attack Strategy

According to the standard Bayesian network inference algorithm, we could get the result that the ability of COG reduces to 78% (22% off) after a time slice behind the end of the second operation.

4 Conclusions and future work

The degradations of COG are different under different attack strategies according to the case study in section 3. So it is hard for staffs to work out the optimal attack strategy without supports, which may lead to an inferior strategy. For example, if staffs are asked to make to choice between the first and second attack strategies, they may not choose the first one (attack the r1 radar continually) with their intuition thinking, because the r1 radar has high defendable ability and strong self repair ability. However, the case study in section 3 shows a contrary result (the first attack strategy is better than the second one), which may leads staffs to think over this question before making decisions.

Applying the MEBN into the COG analysis of SoS is a effective method to solve the target selection problem in combat opponent SoSs. Given different inputs, the analysis model can get different outputs. The comparisons of outputs can be very useful in high opponent combat environments, which support the decision making. However, real operation SoS attack is the effect based operation in multiple phase operations, which seeks the most degradation of COG computed quickly by this method. So our next work will focus on developing the analysis software based on this model with standard Bayesian network interfaces. We also will try to find the effective search algorithm in attack strategies selection process. All this could do help to decisions support in real operation SoS attack scenarios.

5 Acknowledgments

This work is supported in part by National Nature Science Foundation of China (No. 91024006, No. 71031007, No. 71001105) and Postgraduate Innovation Scholarships of National University of Defense Technology (No. S100501). The Bayesian network models described in this paper were created using the GeNIe modeling environment developed by the Decision Systems Laboratory of the University of Pittsburgh (http://www.sis.pitt.edu/~dsl).

6 References

[1] Eisner, H., Marciniak, J., McMillan, R.: Computer-Aided System of Systems (S2) Engineering. In: 1991 IEEE International Conference on Systems, Man, and Cybernetics, pp. 13--16. IEEE Press, Charlottesville, 1991.

[2] Albert, R., Jeong H., Barabtsi, A.L.: Attack and Error Tolerance of Complex Networks. Nature. 406, 378--382, 2000.

[3] Carley, K.M.: Dynamic Network Analysis in Dynamic Social Network Modeling and Analysis: Workshop Summary and Papers. Committee on Human Factors, National Research Council. pp. 133--145, 2003.

[4] Jouni P.: Decision Analytical Approach to Effects-Based Operations. Master Thesis of HELSINKI UNIVERSITY OF TECHNOLOGY. pp. 35--63, 2009.

[5] Lucia Falzon, Using Bayesian network analysis to support centre of gravity analysis in military planning. European Journal of Operational Research. 170, 629--643, 2006.

[6] Kuter, U., Nau, D., Gossink, D., Lemmer, J.F.: Interactive course-of-action planning using causal models. In: Proceedings of the Third International Conference on Knowledge Systems for Coalition Operations (KSCO-2004), pp. 1--12. Czech Republic, 2004.

[7] Laskey, K.B.: MEBN: A Language for First-Order Bayesian Knowledge Bases. Artificial Intelligence. 172(23), 1--51, 2007.

[8] Wright, E., Mahoney, S.M., Laskey, K.B., Takikawa, M., Levitt, T.: Multi-Entity Bayesian Networks for Situation Assessment. In: Proceedings of the Fifth International Conference on Information Fusion, vol. 2, pp. 804--811. Arlington, 2002.

[9] Deng, H.J., Yin, Q.J., Hu, J.W., Cha, Y.B.: Tactical intention recognition based on multi-entity Bayesian network. Systems Engineering and Electronics. 32(11), 2374--2379, 2010.

[10] Mahoney, S.M., Laskey, K.B.: Constructing Situation Specific Networks. In: Proceedings of the Fourteenth Conference of Uncertainty in Artificial Intelligence, pp. 487--498. San Mateo, CA, Morgan Kaufmann, 1998.

Multiple Offer Strategy for Automated Negotiation Agents

Kivanc Ozonat
HP Labs
1501 Page Mill Road
Palo Alto, CA
kivanc.ozonat@hp.com

ABSTRACT

Automated negotiation agents negoatite issues of an e-commerce transaction with human consumers on behalf of e-commerce vendors, and they can increase the financial benefits of the consumer and the vendor jointly. Both the artificial intelligence community and the economics (game-theory) community have proposed methods for negotiation agent design. The focus has been on agents that are restricted to make a single counteroffer to the consumer at every round of the negotiation. Recent studies from the psyhchology community, however, indicate that making multiple counteroffers per negotiation round can be beneficial to both the consumer and the vendor. In light of this, we design an automated software agent that makes multiple offers at every round. We devise a probabilistic strategy that guides the agent to find the optimal set of counteroffers.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

automated, negotiation, multiple offers, statistics

1. INTRODUCTION

Negotiation is the process of reaching an agreement between two (or more) parties on the issues underlying a transaction. Negotiations are an integral part of business life, and are instrumental in reaching agreements among the parties.

Over the last two decades, electronic commerce has become widely adopted, providing consumers with the ability to purchase products and services from businesses online. Electronic commerce offers many advantages, including increased operational efficiency for the businesses, reduction of the inventory costs, and availability of the product and service 24 hours a day. Yet, after two decades, electronic commerce systems still lack the ability to enable the businesses to negotiate with consumers during the purchases. There is a need for automated, online software agents that can negotiate with consumers on behalf of businesses.

Designing automated agents for negotiation has been explored across multiple disciplines, in artificial intelligence [5, 6, 7], human psychology [17, 18, 19, 22] and statistical learning [3, 9]. One form of negotiation that has received much attention in these disciplines is the *bilateral sequential negotiation*. A bilateral sequential negotiation typically starts with one party (e.g., the *buyer*) making an offer on each of the negotiated issues. The issues can include, for instance, the price of a product or service, its delivery time and its quantity. The opposing party (e.g., the *seller*) has three options: (i) it accepts the offer, (ii) it rejects the offer and makes a counter offer on each issue, or (iii) it rejects the offer and terminates the negotiation. The process continues in rounds, where, the buyer and the seller make and respond to counter offers.

The artificial intelligence community has focused on finding game-theoretic solutions to the bilateral sequential negotiation problem. Each of the two parties is assumed to have a *utility function* of the negotiated issues. The utility function measures how preferable an offer is to the party. In the game-theoretic model, each party is assumed to have the goal of maximizing its utility function, and the aim is to find an agreement that is *Pareto-optimal*, i.e., an agreement where the utility of one party cannot be increased any further without decreasing the utility of the other [6].

Many game-theoretic models of negotiation assume that each party knows the opposing party's utility function [6], while more realistic models employ statistical algorithms that *learn* the opposing party's utility function [3, 5, 9]. The learning can be based, for instance, on a training set of issue offers from the parties.

One of the primary goals of an automated agent during the negotiation is to prevent the buyer from terminating the negotiation without reaching an agreement. If the agent's offers are consistently far away from what is acceptable to the buyer, the buyer is likely to walk away from the negotiation. Recent studies from the psychology community indicate that making multiple counteroffers per negotiation round can be beneficial to both the consumer and the vendor [21]. The focus of the game theory and artifical intelligence literature, however, have been on agents that are restricted to make a single counteroffer to the consumer at every round of the negotiation.

We address the question of designing an automated software agent that makes multiple offers at every round. We devise a probabilistic strategy that guides the agent to find the optimal set of counteroffers. Our approach to positioning the multiple offers strategically would be to cluster the issue vectors based on their proximity under some distance d such that the intra-cluster distances are small and the inter-cluster distances are large [10]. Then, from each cluster, we select the cluster member that is closest to the buyer's last offer.

2. MULTIPLE OFFER MODEL

2.1 Assumptions

We assume the following about the buyer's behavior during the negotiation:

1. The agent has the knowledge of (or can estimate from the buyer's history of offers during the negotiation) how close any two offers are from the perspective of the buyer.

2. For any buyer, there exist an "indifference volume" $V(u)$ around each vector u in the issue space, defined as the set of issue vectors where the buyer's gains from preferring any $v \in V(u)$ over u is very small. In particular, the buyer is unlikely to risk termination of the negotiation by insisting on u instead of some other $v \in V(u)$. The agent does not know the size or shape of the volume $V(u)$.

The first assumption is very common in the artificial intelligence view of the negotiation literature. While the game-theoretic approaches assume that the agent makes its counteroffers based on a buyer's utility function known to both parties, the artificial intelligence-based approaches often focus on designing agents that make counteroffers by estimating distances between issue points in the vector space. In [5], for instance, the agent offers the point closest to the buyer's last offer based on some estimated distance.

The second assumption follows from that the buyer is unlikely to form mathematical functions (e.g., utility functions, iso-curves, etc.) upon which its decisions are based. Instead, the buyer is more likely to make its decisions based on a mixture of quantitative and qualitative criteria. We note that we only assume the existence of an indifference volume around each issue vector. We do not make any assumptions about the size or the shape of the volume.

2.2 Model

In designing negotiation agents in the artificial intelligence community, a very common offer strategy is to offer the issue, which is closest to the buyer's last offer among a set of issue vectors. Often, the agent forms the set such that all issue vectors in the set has the same utility for the agent, i.e., the issue vectors are on the same agent utility *isocurve*. Then, the agent offers the member that is closest to the buyer's last offer.

In this strategy, the distance between an issue vector u and the buyer's last offer y is often measured as a weighted sum of their elementwise distances, i.e.,

$$D(u, y) = \sum_i w_i D_i(u_i, y_i), \tag{1}$$

where the u_i and y_i are the i^{th} elements of the vectors u and y, and the weights w_i are computed based on the estimated preferences of the buyer. The agent can estimate these preferences from the buyer's offer history. The distance measure D_i can be, for instance, the elementwise absolute difference ($|u_i - y_i|$) between the two vectors.

What if the agent is to make two offers at each round? Extending the approach in [5], the agent can select the closest two issue vectors to y on the agent's isocurve as offers. However, this is not necessarily a good strategy especially if these two offers are very close to each other. If the two closest vectors are within a close neighborhood of each other, for instance, this strategy would not add much beyond offering just one of these two vectors.

At the other extreme is to select the first offer as the closest one to y on the isocurve, and the second offer on the isocurve as far from the first offer as possible. This strategy is far from optimal since the second offer takes into account neither the buyer's offer y nor his/her preferences.

To quantify the discussion, assume the buyer offers y in the last round. Let $p_y(u_1)$ denote the probability that the buyer terminates the negotiation in response to agent's offer u_1, and let $p_y(u_2|u_1)$ denote the conditional probability that the buyer terminates the negotiation in response to the offer u_2 given the buyer would terminate it in response to u_1. Then, by the definition of conditional probability, the probability of the buyer continuing with the negotiation is

$$1 - p_y(u_1, u_2) = 1 - p_y(u_1)p_y(u_2|u_1). \tag{2}$$

How to model $p_y(u_1)$ and $p_y(u_2|u_1)$? From the discussion in [2002] and [2006], one viable option is to let $p_y(u_1)$ to be proportional to the distance between u_1 and y, i.e.,

$$p_y(u_1) = k_1|y - u_1|, \tag{3}$$

for some constant k_1.

What about $p_y(u_2|u_1)$, i.e., the probability that u_2 gets rejected given that u_1 would be rejected? Intuitively, this conditional probability should depend on both the distance between y and u_2, and the distance between u_1 and u_2. In particular, if the distance between u_1 and u_2 is small, the conditional probability should depend more on the distance between u_1 and u_2. On the other hand, if the distance between u_1 and u_2 is large, the conditional probability should be more dependent on the distance between y and u_2. Accordingly, we let

$$p_y(u_2|u_1) = \begin{cases} k_2|y - u_2| & \text{if } |u_1 - u_2| \geq \epsilon, \\ 1 & \text{if } |u_1 - u_2| < \epsilon, \end{cases} \tag{4}$$

for some constants k_2 and ϵ.

Then, the probability that the buyer will continue with the negotiation becomes

$$1 - p_y(u_1, u_2) = \begin{cases} 1 - k_1k_2|y - u_1||y - u_2| & \text{if } |u_1 - u_2| \geq \epsilon, \\ 1 - k_1|y - u_1| & \text{if } |u_1 - u_2| < \epsilon. \end{cases} \tag{5}$$

We note that the probability $k|y - u| < 1$ for any k_1,k_2 and u since p_y is a probability mass function.

If u_2 is placed very close to u_1 (i.e., within a distance of ϵ), then the acceptance probability is guaranteed to be lower than the case with u_2 placed far away from u_1 (i.e., outside a distance of ϵ). This follows from

$$1 - k_1k_2|y - u_1||y - u_2| > 1 - k_1|y - u_1|, \tag{6}$$

since $k|y - u| < 1$ for any k_1,k_2 and u.

One can then extend the probability of continuing with the negotiation to $N > 2$ offers, and show that with N simulatenous agent offers, the probability of the buyer walking away from the negotiation is minimized if no two offers are

within a distance of ϵ. In this case, the probability of the buyer walking away from the negotiation is

$$\prod_i k_i|y - u_i|, \tag{7}$$

provided no two offers are within a distance of ϵ of each other.

Minimizing (8) is equivalent to minimizing

$$\sum_i \log k_i + \log |y - u_i|. \tag{8}$$

The optimization problem in (8) is, in general, intractable, and thus we use a suboptimal strategy to minimize (8).

Our multiple offer strategy is to first partition the issue space using graph partitioning [10] into multiple clusters such that the partitioning satisfies the following conditions:

1. Any two issue vectors, u and v, within the same clusters are (very likely to be) close to each other, i.e., within a distance ϵ.

2. Any two issue vectors, u and v, between two clusters are (very likely to be) far from each other, i.e., outside a distance of ϵ.

Once the partitioning stage is done, we select one issue vector from each cluster as an offer. In each cluster, we select the offer that is closest to the last offer of the buyer. This strategy ensures that the offer closest to the buyer's last offer in the issue space is always selected. Thus, the first offer is the same as that one would get from the strategies in [5]. It further ensures that the remaining offers are not very close.

The motivation for the multiple offer strategy is that it is rare for the agent to have perfect estimates of the buyer's preferences w_i. The inaccuracies in estimating the preferences will impact the distance function. For most negotiation scenarios, it is reasonable to assume that D_i would be the Euclidean distance or the city-block distance. In fact, some prior work just assume that D_i is simply the absolute elementwise difference between vectors (i.e., the city-block distance).

3. CLUSTERING

3.1 Definitions

Let V denote a finite set of elements, and E be a set of edges e such that each edge is connects of the elements in V. $G = (V, E)$ is called a graph with the vertex set V, and edge set E. A weighted graph is a graph that has a positive number $w(e)$ associated with each edge e, called the weight of edge e. Denote a weighted graph by $G = (V, E, w)$. An edge e is said to be incident with a vertex u when e connects u to another edge.

We say that there is a path between vertices v_1 and v_k when there is an alternative sequence of distinct vertices and edges v_1, e_1, v_2, e_2,..., e_{k-1}, v_k such that $v_i, v_{i+1} \in e_i$, for $1 \leq i \leq k-1$. A graph is connected if there is a path for every pair of vertices. We assume that the graphs in this work are connected.

3.2 Graph clustering

The graph clustering problem with k clusters is formalized as follows [10]: Given a graph $G = (V, E)$, find the clustering V_1, V_2,..., V_k that minimizes the number of edges of E whose incident vertices belong to different subsets, provided

- $V_i \cap V_j = \emptyset$ for $i \neq j$,
- $|V_i| = |V|/k$, and $\cup_i V_i = V$,

If the edges have weights associated with them, the graph clustering problem can be extended to account for the edge weights. In this case, the problem can be formulated as minimizing the sum of the edge weights belonging to different subsets provided the two conditions listed above. Given a clustering, the number of edges whose incident vertices belong to different subsets is called the edge-cut of the clustering.

The graph G can be clustering using a multi-level algorithm. The graph G is first coarsened down to a few hundred vertices, a partitioning of this smaller graph into k clusters is computed, and then this partition is projected back towards the original graph (finer graph). At each step of the graph uncoarsening, the partition is further refined. The refinements reduce the edge-cut since Since the finer graph has more degrees of freedom.

Consider a weighted graph $G_0 = (V_0, E_0)$ with weights both on the edges. A multi-level graph clustering algorithm consists of the following three stages:

- Coarsening Phase: The graph G_0 is transformed into a sequence of smaller graphs G_1, G_2,..., G_m such that $V_0 > V_1 > V_2 > .... > V_m$.
- Partitioning Phase A 2-way partition P_m of the graph $G_m = (V_m, E_m)$ is computed that partitions V_m into two parts, each containing half the vertices of G_0.
- Uncoarsening Phase The partition P_m of G_m is projected back to G_0 by going through intermediate partitions $P_{m-1}, P_{m-2}, ..., P_1, P_0$.

3.3 Matching

A matching of a graph $G_i = (V_i, E_i)$ is a set of edges such that no two edges are incident on the same vertex. The coarser graph G_{i+1} is constructed from G_i by finding a matching of G_i and collapsing the matched vertices into multinodes. The unmatched vertices are simply copied over to G_{i+1}. The matching of a graph is obtained through forming *maximal matchings*. A matching is maximal if any edge in the graph that is not in the matching has at least one of its endpoints matched.

A maximal matching can be generated efficiently using a randomized algorithm as follows:

- Vist the vertices in random order.
- If a vertex u has not been matched yet, then randomly select one of its unmatched adjacent vertices.
- If such a vertex v exists, include the edge (u, v) in the matching and mark vertices u and v as being matched.
- If there is no unmatched adjacent vertex v, then vertex u remains unmatched in the random matching.

The complexity of the above algorithm is $O(E)$.

While random matching is an efficent technique to obtain a maximal matching, it does not target minimizing the edge-cut. Consider a graph $G_i = (V_i, E_i)$, a matching M_i that is used to coarsen G_i , and its coarser graph $G_{i+1} = (V_{i+1}, E_{i+1})$ induced by M_i. If A is a set of edges, define $W(A)$ to be the sum of the weights of the edges in A. It can be shown that

$$W(E_{i+1}) = W(E_i) - W(M_i). \quad (9)$$

Thus, the total edge-weight of the coarser graph is reduced by the weight of the matching. Hence, by selecting a maximal matching Mi whose edges have a large weight, we can decrease the edge-weight of the coarser graph by a greater amount. As the analysis in [27] shows, since the coarser graph has smaller edge-weight, it also has a smaller edge-cut. Finding a maximal matching that contains edges with large weight is the idea behind the heavy-edge matching. A heavy-edge matching is computed using a randomized algorithm similar to that for computing a random matching described earlier. The vertices are again visited in random order. However, instead of randomly matching a vertex u with one of its adjacent unmatched vertices, we match u with the vertex v such that the weight of the edge (u, v) is maximum over all valid incident edges (heavier edge).

3.4 Partitioning

The second phase of a multilevel algorithm computes a high-quality bisection (i.e., small edge-cut) P_m of the coarse graph $G_m = (V_m, E_m)$ such that each part contains roughly half of the vertex weight of the original graph. Since during coarsening, the weights of the vertices and edges of the coarser graph were set to reflect the weights of the vertices and edges of the finer graph, G_m contains sufficient information to intelligently enforce the balanced partition and the small edge-cut requirements.

The Kernighan-Lin algorithm [31] is iterative in nature. It consists of the following iterations:

1. Start with an initial partition of the graph.
2. Search for a subset of vertices from each part of the graph such that swapping them leads to a partition with a smaller edge-cut.
3. Stop if you cannot find such two subsets. This indicates that the partition is at a local minimum.

Each iteration of the KL algorithm described in [31] takes $O(ElogE)$ time.

The Kernighan-Lin algorithm finds locally optimal partitions when it starts with a good initial partition and when the average degree of the graph is large [4]. If no good initial partition is known, the KL algorithm is repeated with different randomly selected initial partitions, and the one that yields the smallest edge-cut is selected.

Suppose P is the initial partition of the vertices of $G_m = (V_m, E_m)$. The gain g_v, of a vertex v is defined as the reduction on the edge-cut if vertex v moves from one partition to the other. This gain is given by

$$g_v = \sum_{(v,u)\in E \cap P[v]\neq P[u]} w(u,v) - \sum_{(v,u)\in E \cap P[v]=P[u]} w(u,v) \quad (10)$$

where $w(v, u)$ is weight of edge (v, u). If g_v is positive, then by moving v to the other partition the edge-cut decreases by g_v, whereas if g_v is negative, the edge-cut increases by the same amount. If a vertex v is moved from one partition to the other, then the gains of the vertices adjacent to v may change. Thus, after moving a vertex, we need to update the gains of its adjacent vertices.

Given this definition of gain, the KL algorithm then proceeds by repeatedly selecting from the larger part a vertex v with the largest gain and moves it to the other part. After moving v, v is marked so it will not be considered again in the same iteration, and the gains of the vertices adjacent to v are updated to reflect the change in the partition. The original KL algorithm [9], continues moving vertices between the partitions, until all the vertices have been moved.

3.5 Refinement

During the uncoarsening phase, the partition P_m of the coarser graph G_m is projected back to the original graph, by going through the graphs G_{m-1}, G_{m-2},...., G_1. Since each vertex of G_{i+1} contains a distinct subset of vertices of G_i, obtaining P_i from P_{i+1} is done by simply assigning the set of vertices V_i^v collapsed to $v \in G_{i+1}$ to the partition $P_{i+1}[v]$.

Even though P_{i+1} is a local minimum partition of G_{i+1}, the projected partition P_i may not be at a local minimum with respect to G_i. Since G_i is finer, it has more degrees of freedom that can be used to improve P_i, and decrease the edge-cut. Hence, it may still be possible to improve the projected partition of G_{i-1} by local refinement heuristics. For this reason, after projecting a partition, a partition refinement algorithm is used. The basic purpose of a partition refinement algorithm is to select two subsets of vertices, one from each part such that when swapped the resulting partition has a smaller edge-cut.

The refinement algorithm consists of the following steps:

1. Use the projected partition of G_{i+1} onto G_i as the initial partition. (Note that the projected partition is already a good partition).
2. Apply vertex swaps to decrease the edge-cut.

$$g_v = \sum_{(v,u)\in E \cap P[v]\neq P[u]} w(u,v) - \sum_{(v,u)\in E \cap P[v]=P[u]} w(u,v) \quad (11)$$

3. Terminate the algorithm when no further decrease in the edge-cut can be made through edge swaps.

4. SIMULATIONS

Automated software agents for bilateral negotiations can be designed through either game-theoretic methods or statistical learning approaches. In either case, a robust design, evaluation and testing of the agents requires the existence of a (training) set of issue offers representing typical behaviors of the opposing party. When a set of issue offers recorded during an actual negotiation process is available, it can be used to design, evaluate and test the agents. However, it is very rare to have recordings of actual negotiations, in which case, the issue offers need to be obtained through simulations.

In [5], the buyers' negotiation behaviors have been modeled through a sequence of functions; each function imitates the buyer's offers in the issue space. The functions are monotonic in the issue values and belong to families of parametrized functions. The models in [5] assume a single-issue negotiation scenario, which we discuss in section 4.1. We then extend it to the scenario with multiple negotiated issues in section 4.2.

4.1 Single-issue buyer profiles

According to [5], negotiation tactics can be classified into two main categories: (i) time-dependent (or resource-dependent) tactics, and (ii) behavior-dependent tactics. Time-dependent tactics determine how the issue offer $y_{n,d}$ changes as a function of time n. Resource-dependent tactics are a generalization of the time-dependent tactics, where resources other than time (e.g., inventories) are considered. The behavior-dependent tactics, on the other hand, determine how the issue offer $y_{n,d}$ changes as a function of the counter offers of the opposing party.

We focus on the time-dependent tactics. In [5], the evolution of the offer $y_{n,d}$ through time is given as

$$y_{n,d} = y_{\min,d} + \alpha_d(n)(y_{\max,d} - y_{\min,d}) \tag{12}$$

where $[y_{\min,d}\ y_{\max,d}]$ is the range of values for issue d, and α_d is a time-varying function.

There are many choices for the function α_d as discussed in [5]. Each function is monotonic in the issue value, and has five parameters: β_d, $t_{\max}$, $y_{\min,d}$, $y_{\max,d}$ and κ. The parameter β_d controls the convexity of the function. When $\beta_d < 1$, the function is convex, and when $\beta_d > 1$ the function is concave. The following is one such example function:

$$\alpha_d(n) = \kappa + (1-\kappa)(\min(n, t_{\max})/t_{\max})^{1/\beta_d}. \tag{13}$$

Of the five parameters, κ is a constant (usually set to 0), β_d controls the concession behavior, and $t_{\max}$ is the maximum number of negotiation rounds. The notation $\min(n, t_{\max})$ implies the smaller of n and $t_{\max}$. As β_d is decreased, the party makes more "boulware" offers, i.e., the party stays closer to the $y_{\min,d}$ value until the time is almost exhausted, whereupon it concedes up to $y_{\max,d}$. As β_d is increased, the party makes more "conceder" offers, i.e., the party quickly moves up to $y_{\max,d}$.

4.2 Multi-issue buyer profiles

The treatment in [5] of the negotiation tactics focuses on single-issue negotiation settings. When the negotiation involves multiple issues, however, one needs to take into account the dependencies among the issues.

The model in [5] assumes that the negotiating party (e.g., the buyer) starts with an initial issue offer $y_{\min,d}$, and increases the issue value at each time step (or round) n, according to, for instance (12) and (13). The multiple issue negotiation literature, on the other hand, focuses on game-theoretic approaches with utility functions. The assumption is that a negotiating party starts with an initial issue vector with utility equal to $U_{initial}$ and aims to reach an agreement at a target vector with utility U_{target}. By the very nature of negotiation, U_{target} is less than $U_{initial}$. The literature also takes the view that the party decreases the utility value monotonically at every round.

To extend the model in [5] for simulating buyer's behavior in a multi-issue setting, we randomly select one of the buyer's target vectors as $y_{\max,d}$. We then randomly select another vector as $y_{\min,d}$, such that each element of $y_{\max,d}$ is greater than or equal to the corresponding element of $y_{\min,d}$. This condition is needed to ensure that the monotonic functions (e.g., (13)) can be used to move from $y_{\min,d}$ to $y_{\max,d}$ in a finite number of rounds. At each time n, we either keep the β_d value for each issue d (no change in the state), or switch to a new β_d value for each issue d according to some prior transition probability as discussed in [5].

The parameter β_d needs to be selected to ensure the convexity (or concavity) of α_d. According to the Faratin's approach, the negotiator's tactic can be characterized as either "boulware" or "conceder". In particular, for $\beta_d < 1$, the tactic is boulware, while for $\beta_d > 1$, we observe the conceder tactics. The function (13) is convex in n, and the degree of convexity, determined by the value of β_d, identifies the tactic. As β_d is increased, the offers get more conciliatory, and as β_d is decreased, the offers become more boulware. Faratin provides functions alternative to that in (13) (e.g., an exponential function instead of a polynomial function), yet their common characteristic is that they are convex in n, and their degree of convexity is determined through the parameter β_d.

Since a function is (strictly) convex if and only if its second derivative is (strictly) non-negative, and that a weighted sum of second derivatives is equal to the second derivative of the weighted sums, it follows that the additive utility function, U, defined as

$$U = \sum_d w_d U_d \tag{14}$$

for some weights $w_d > 0$ and the issue utilities U_d, is guaranteed to be convex as long as $\beta_d < 1$ for each d, and is guaranteed to be concave as long as $\beta_d > 1$ for each d. Thus, in transitioning from one set of β_d values to another in the simulations, we make sure that, in each state, either $\beta_d < 1$ for each d or $\beta_d > 1$ for each d. This ensures the convexity (or the concavity) in the utility space as well as in the issue space.

4.3 Results

We evaluate the performance of the seller's agent. The agent negotiates issues with buyers on behalf the seller, where the buyer's offers on the issues are generated through the simulations described in sections 4.1 and 4.2. The agent is available to it a rank-ordered list of the issue vectors with the higher-ranking vectors being more advantageous to the seller. The top 20 percent of the issue vectors in the list are acceptable to the seller. The buyer has a different rank-ordered list of the issue vectors, and the top 20 percent of the issue vectors in the list are acceptable to the buyer.

In Fig. 1, we assigned a *utility score* (in the 0-100 range) to each issue point with higher-ranking points (in the buyer's ranked-ordered list) with higher utility scores, and computed the mean of the utility values of the agent's counter offers. In Fig. 1, the concession behavior of the buyer is that of a conceder (i.e., $\beta > 1$). As the number of offers N by the agent increases, the mean utility value for the buyer increases. The gains are greatest when N is increased from 2 to 5, and the gains are small when N is further increased to 10. This might indicate that generating 5 offers per negoti-

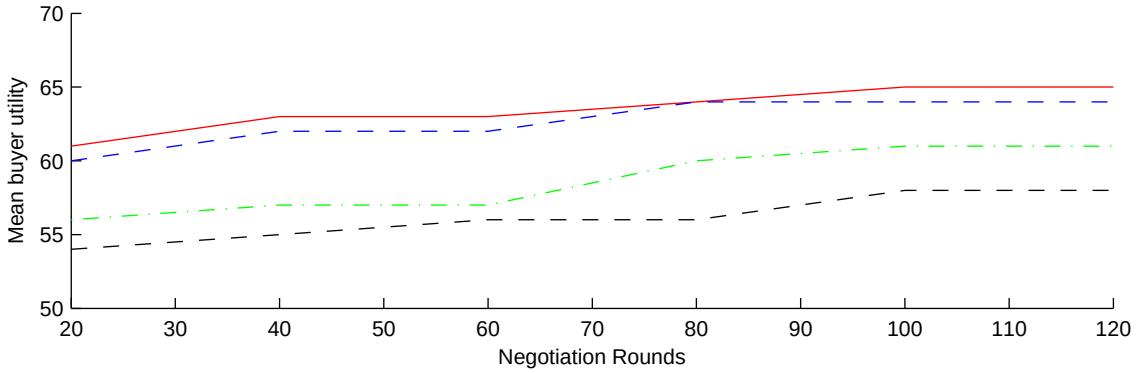


Figure 1: The buyer's utility as a function of the number of negotiation rounds. Conceder behavior. Single offer (black dashed). Two offers (green dash dot). Five offers (blue dashed). Ten offers (red solid).

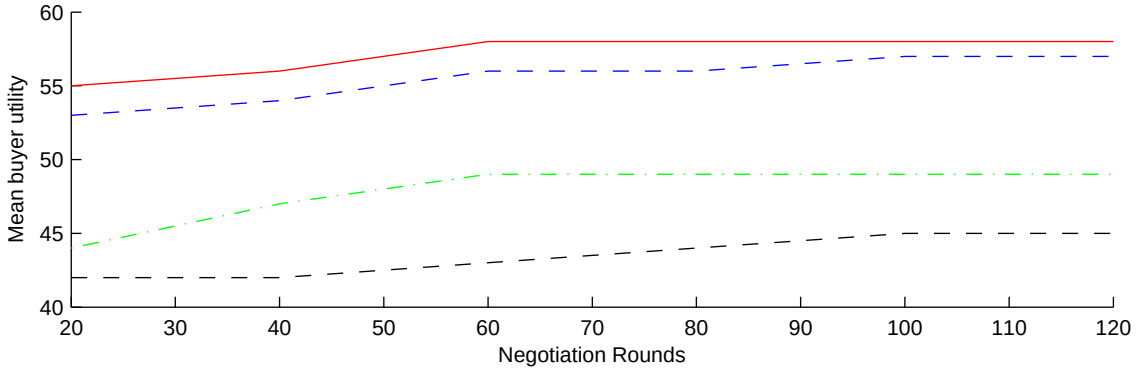


Figure 2: The buyer's utility as a function of the number of negotiation rounds. Boulware behavior. Single offer (black dashed). Two offers (green dash dot). Five offers (blue dashed). Ten offers (red solid).

ation round is a sufficiently good strategy.

Fig. 2 shows the mean of the utility values of the agent's counter offers for a boulware buyer model (i.e., $\beta < 1$). Similar to the number of offers by the agent increases, the mean utility value for the buyer increases. However, the gains from the multiple offers in this case are higher, indicating that the multiple offer strategy is more benefitial for boulware buyers than it is for conceder buyers. Finally, Fig. 3 shows the behavior for a linear buyer model. Similar to Fig. 1 and Fig. 2, the mean utility value of the buyer increases as the number of offers by the agent increases.

5. CONCLUSIONS

Automated negotiation agents negoatite issues of an e-commerce transaction with human customers on behalf of e-commerce vendors. The agents proposed by the artificial intelligence and game theory communities have focused on designing agents that make a single offer to the customer at every round of the negotiation. Recent studies from the psyhchology community, however, indicate that making multiple counteroffers per negotiation round can be beneficial to both the consumer and the vendor. In this work, we designed automated agents that make multiple offers to the customer at every negotiation round. We have utilized graph-based statistical clustering to partition the space of the offers and generate the multiple offers. Our results, based on popular buyer behavior models, indicate that this strategy leads to a significant increase in the customer's utility without decreasing the agent's utility.

References

[1] A. Barron, J. Rissanen, and B. Yu. The minimum description length principle in coding and modeling. *IEEE Transactions in Information Theory*, 44(6):2743-2765, 1998.

[2] M. Beal, Z. Ghahramani, and C.E. Rasmussen. The infinite hidden Markov model. In *NIPS*, 2002.

[3] R.M. Coehoorn and N. Jennings. Learning an opponent's preferences to make effective multi-issue negotiation trade-offs. In *ICEC*, 2004.

[4] T. Cover and J. Thomas. *Elements of Information Theory*, New York, Wiley, 1991.

[5] P. Faratin, C. Sierra, and N. R. Jennings. Negotiation decision functions for autonomous agents. *Int. Journal of Robotics and Au- tonomous Systems*, 24(3-4):159Ű182, 1998.

[6] S. Fatima, M. Wooldridge, and N. R. Jennings. Multi-issue negotiation under time constraints. In *AAMAS*, 2002.

[7] S. Fatima, M. Wooldridge, and N. R. Jennings. An agenda based framework for multi-issues negotiation. *Artificial Intelligence Journal*, 152(1):1Ű45, 2004.

[8] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky. An HDP-HMM for systems with state persistence. In *ICML*, 2008.

[9] T. Ito, H. Hattori, and M. Klein. Multi-issue negotiation protocol for agents: Exploring nonlinear utility spaces. In *IJCAI*,

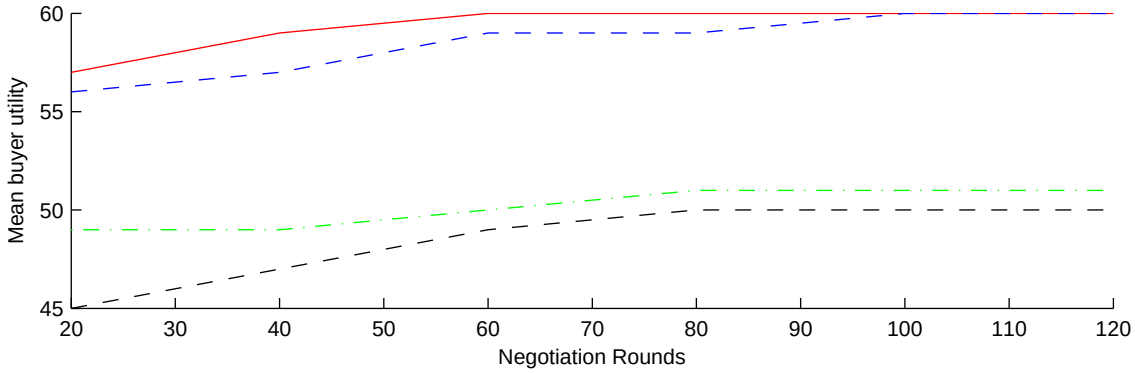


Figure 3: The buyer's utility as a function of the number of negotiation rounds. Linear behavior. Single offer (black dashed). Two offers (green dash dot). Five offers (blue dashed). Ten offers (red solid).

2007.

[10] G. Karypis and V. Kumar. A coarse-grain parallel formulation of multilevel k-way graph partitioning algorithm. *Siam Conference on Parallel Processing for Scientific Computing*, 2004.

[11] A. N. Kolmogorov. Three approaches to the quantitative definition of information. *Probl. Inform. Transm.* vol. 1. pp. 4-7, 1965

[12] V. Pavlovic, J. M. Rehg, T. J. Cham, and K. P. Murphy, A dynamic bayesian network approach to figure tracking using learned dynamic models. In *Intl. Conf. Computer Vision*, 1999.

[13] V. Pavlovic, J. Rehg, and J. MacCormick. Learning switching linear models of human motion. In *NIPS*, 2001.

[14] C. E. Shannon and W. Weaver. *The mathematical theory of communication.* University of Illinois Press, Urbana, IL, 1949.

[15] R. J. Solomonoff. A formal theory of inductive inference I,II, *Inform. and Control*, vol. 7 pp. 1-22, 224-254, 1964.

[16] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of American Statistics Association*, 101(476):1566-1581, 2006.

[17] G. A. Van Kleef, C. K. W. De Dreu, and A. S. R. Manstead. The interpersonal effects of anger and happiness in negotiations. *Journal of Personality and Social Psychology*, 86, pp. 57Ű76, 2004.

[18] G. A. Van Kleef, C. K. W. De Dreu, and A. S. R. Manstead. The interpersonal effects of emotions in negotiations: A motivated information processing approach. *Journal of Personality and Social Psychology*, 87, pp. 510-528, 2004.

[19] R. Vetschera. Preference structures and negotiator behavior in electronic negotiations. *Decision Support Systems*, vol. 44, pp. 135-146, 2007.

[20] C. Wallace and D. L. Dowe. Minimum mesaage length and Kolmogorov complexity. *The Computer Journal*, vol. 42, no. 4, pp. 270-283, 1999.

[21] Y. Yang and S. Singhal.Designing an intelligent agent that negotiates tactfully with human counterparts: A conceptual analysis and modeling framework. *Proceedings of HICSS*, 2009.

[22] G. A. Yukl. Effects of situational variables and opponent concessions on a bargainerŠs perception, aspirations, and concessions. *Journal of Personality and Social Psychology*, vol. 29, pp. 227-236, 1974.

Artificial Intelligence Techniques for Understanding Gothic Cathedrals

Stefaan Van Liefferinge[1], Charles Hollingsworth[2], Rebecca A. Smith[1], Michael A. Covington[2], and Walter D. Potter[2]
[1]Lamar Dodd School of Art, The University of Georgia, Athens, Georgia 30602, USA
[2]Institute for Artificial Intelligence, The University of Georgia, Athens, Georgia 30602, USA

Abstract— *This paper introduces a work in progress on a new research topic for artificial intelligence. It presents the first stages of research that investigates the capacities of AI to understand the architectural rules that define historic architecture: specifically those of Gothic cathedrals. Gothic architecture has its own logic. It follows rules that are defined by physical and structural constraints as well as by the style of the architecture. This project will analyze specialized descriptions of Gothic architecture and translate them into computer code. On the long run this will result in an intelligent computational model, a program that can understand the structure of a Gothic cathedral and reason about its architecture. This paper introduces the background of this project, sketches a small knowledge base illustrating the approach, and describes the potential impact of this type of research on the study of historic architecture.*

Keywords: Gothic Cathedrals, Architectural Descriptions, Knowledge Representation, Text Understanding, Logic, Computational Reasoning

1. Introduction and overall method

The limits prescribed by architectural design make built structures a good subject for AI experimentation.[1] It is thus not surprising that a number of projects have brought architecture within the scope of AI. For example, architectural historians have indicated how patterns in the language of architecture can be approached as if it were code [1]. Also engineers have investigated how to use AI for architectural research and design. Some have proposed systems to obtain the logical configuration of built structures [2]. Others have developed algorithms for producing computer-generated models of historic architecture [3], [4]. These initiatives underscore the affinity between AI and architecture. However, up to now, research has focused on the buildings themselves without addressing an intermediate medium as, for example, written architectural descriptions.

[1]This research benefited from the generous support of a Digital Humanities Start-Up Level I Grant from the National Endowment for the Humanities (Grant Number HD5110110), a University of Georgia Research Foundation Grant, and from The University of Georgia President's Venture Fund.

This research project involves a method that utilizes a translation of architecture into a logical system. More precisely, it takes advantage of the existence of architectural descriptions in the literature, descriptions that are expressed in a natural language and that follow specific rules. Indeed, the conventions which architecture generally obeys have defined a language, a structured and logical means of describing architecture. Architectural historians commonly manipulate this tool to communicate their analysis of buildings. These descriptions reflect the structure and rules of architecture, and, conversely, the conventions that architectural texts follow are well adapted to describing built structures.

Thus, instead of searching for a direct mapping between actual buildings and an intelligent computer system, our research investigates the application of AI technologies for understanding written descriptions of buildings. For this task a subset of architecture, Gothic cathedrals, was chosen [5]. This subset is well adapted because Gothic cathedrals are known for being organized following recurrent patterns. Their internal organization is valuable and serves as a heuristic to codify the logic of their architecture.

While describing architecture is a practice in obeying conventions, it is nonetheless a human activity that combines presupposed knowledge, intelligence, and creativity. These factors put this research project well beyond coding or modeling. It searches to catch the processes involved in a specific discipline - architectural history - and to bring these within the reach of computational logic. Ultimately, we would like to implement a system that can automatically translate between English and a computer, and which can reason about any description of architecture.

In broad lines, the adopted method of research is as follows. Schemes commonly used to describe Gothic cathedrals are analyzed and categorized. This classification is used to produce an architectural description language (ADL), a simple subset of English that is sufficient for describing the architectural features of interest. In this ADL we write a generic description that captures those features common to most Gothic cathedrals. Descriptions of particular cathedrals need only describe how they differ from this canonical description, as described in [6]. However, the translation of architectural descriptions into formal logic is no simple

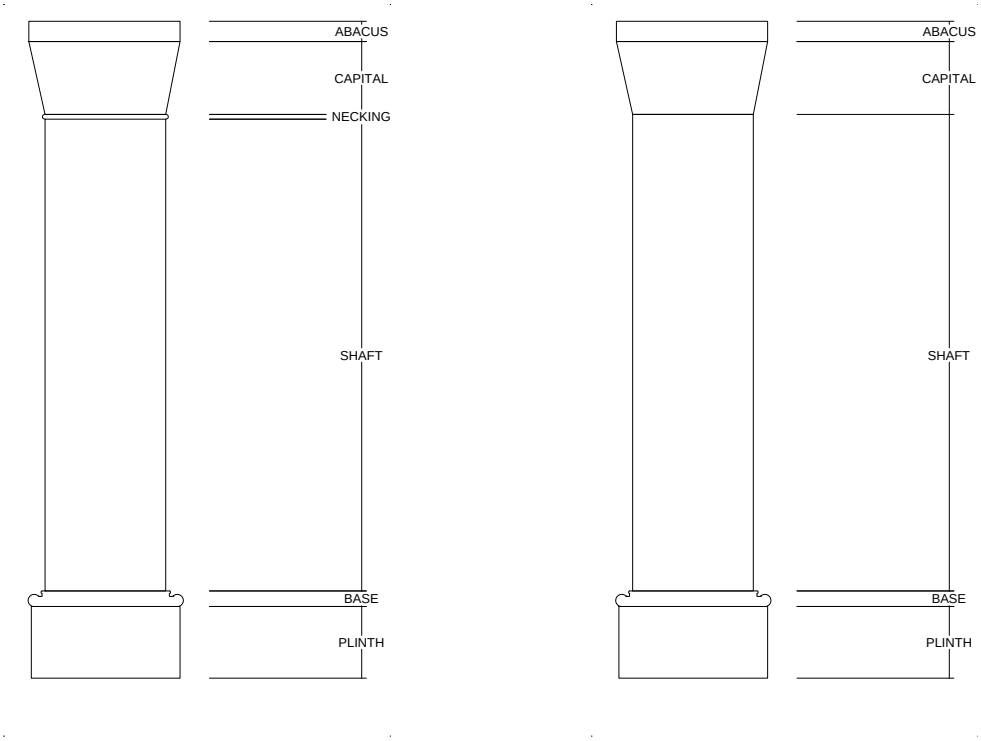


Fig. 1: Column, with and without necking

```
shaft([shaft_inst(1),X]) :- column(X).
has(X, [shaft_inst(1),X]) :- column(X).
```

Fig. 2: Prolog translation of "Every column has a shaft."

```
immediately_above(X,Y) :-
    necking(X),
    shaft(Y),
    has(ParticularColumn, X),
    has(ParticularColumn, Y).
```

Fig. 3: Prolog translation of "If a column has a necking, it is immediately above the shaft."

task. For this, at this stage, we are experimenting with the translation of a small subset of architecture, that of a column, into a declarative language and testing how a computer can operate and interactively reason with this subset.

2. A Minimal Case Study

Because the project aims at identifying knowledge presupposed in descriptions of architecture, knowledge representation is of central importance to our project. Architectural descriptions must be translated into a formal, machine-readable format that can be used for responding to queries and drawing inferences. We decided to render these descriptions as source code for the logic programming language Prolog, because the Prolog interpreter already incorporates question answering and inference. Additionally, much work has already been done on translating from natural language to Prolog [7], [8].

We chose first to devise a knowledge base for describing Gothic columns, with the intention of expanding this later to include other architectural features. Columns were chosen because, while they play a fundamental role in a cathedral, they are fairly simple structures. Furthermore, the features in a column are arranged very simply, one on top of the other, and the order does not vary. However, certain features, such as the base, capital, and shaft, are found in all columns, while others, such as neckings and plinths, are not always present (Fig. 1). We thus needed to be able to represent concepts such as "Every column has a shaft" and "If a column has a necking, then the necking is immediately below the capital." Finally, we needed to account for the fact that many architectural features are repeated, so that one might encounter statements such as "Every vaulting unit has four columns." For this last requirement, we made extensive use

of skolemization [9], a well-known method for eliminating existential quantifiers. The use of skolemization in logical programming is covered in [7], [10]; our implementation is a modified version of that described in [8].

The Prolog translation of the assertion that "Every column has a shaft" is in Figure 2. The Prolog term [shaft_inst(1),X] is a Skolem function, a way of providing a unique identifier for each shaft that is a function of the name of the column of which it is a component. In other words, we are asserting two things: for each column X, there exists a unique shaft named [shaft_inst(1),X]; and column X has that shaft as one of its components. These identifiers are admittedly unwieldy, but should only be used internally by our program: the end user should never have to type them as part of a query. The listing in Figure 3 illustrates how to say "The necking is immediately above the shaft," while bearing in mind that not all columns will have a necking.

This states that, if some particular column has both a necking and a shaft, the necking is immediately above the shaft. Elsewhere we deal with columns without neckings, in which case it is the capital which is immediately above the shaft. This is in fact the default, and if we simply mention a column without specifying whether it has a necking, the system assumes it has none. However, if we subsequently assert that the column does have a necking, the previously derived facts about the relative locations of the capital and shaft must be revised. To do this we make use of nonmonotonic reasoning [11], [12], a form of reasoning that allows previously deduced assertions to be defeated by new evidence. Our implementation is similar to that presented in [13].

At present, here is a very basic set of queries which the user can make to our knowledge base. To get a list of constituent parts of a column named column1, for example, the user can type

```
?- has(column1, X).
```

into the Prolog interpreter. For an exhaustive list of the components of vaulting unit v1 and their various subcomponents, type:

```
?- has(v1,X), has(X,Y).
```

To get a list of all the components below the capital in column1, type:

```
?- has(column1, X), capital(X), above(X, Y).
```

While this simple example already captures some essential capabilities needed for architectural description, we will expand this knowledge base to deal with more and more sophisticated concepts. Eventually, a natural language front end will take the place of the Prolog query interface, making the system more usable, and paving the way for eventually being able to handle more typical architectural texts.

3. Future Impact

Both architectural history and artificial intelligence will benefit from the combination of technology and traditional methods of building analysis. It will allow architectural historians to understand better how we write architectural descriptions, will be a beneficial study tool for students and professionals alike, and will eventually be able to give us accurate visuals of buildings. Engineers working in artificial intelligence can expect from this project an entirely new scope for natural language processing, and to apply artificial intelligence to a previously little-explored field.

Writing for architectural history is a complex and often subjective process. The number of assumptions made when writing a description of a building, particularly when writing about Gothic architecture, is vast. A program that could "read" and analyze architectural descriptions would be entirely new. Such a program would be able to detect trends among various essays and treatises that scholars are often unaware exist. These trends could be as simple as the order in which a building elevation is described, to something as complex as the nature of the geometrical pattern from which the plan is derived. The discovery of patterns within written text could further the study of natural language processing by highlighting how humans describe buildings. It could show how we translate the experience of a building into written description – and possibly underscore the limitations within writing. Previously unknown peculiarities within individual buildings could be found and studied by experts, without affect from the voice of a particular author or the subjective analysis of future scholars.

One of the long-term goals of this project is to create a program that can generate models that integrate 3D images of buildings, expert knowledge of architectural historians, and archaeological data. This can have a profound effect on the study of architectural history and provide a fascinating field of exploration for artificial intelligence. For example, by using written sources, the system could create a building design or plan that is historically more valid than that of a human scholar drawing by CAD or by hand. Applications of this type of image creation program should also be of interest to engineers applying AI methods in the virtual game design industry. Engineers could create buildings and cities that were previously limited to the human imagination. The program could use historic written accounts to understand better what the original author actually meant to describe, even if the account is mistaken or lacking. This will allow us to create models and imagery of buildings that exist only in descriptive accounts.

Often drawings of buildings that no longer exist or may have never existed are inaccurate and misleading. This project opens a door to the generation in the future of accurate and detailed elevations, floor plans, and virtual tours. Such models could, for example, inform preservationists

about previous structural interventions in buildings. This would enable more accurate repairs, which would lead to the continuing survival of many of the world's treasures. The cooperation between these two seemingly remote fields will be mutually beneficial, and likely foreshadows the coming of new interdisciplinary studies – combining artificial intelligence with both architectural history and other fields in the humanities and sciences.

References

[1] William J. Mitchell. *The logic of architecture: Design, computation, and cognition*. MIT Press, Cambridge, MA, USA, 1990.

[2] Atsushi Saito, Junzo Munemoto, and Daisuke Matsushita. Acquiring Configuration Rules of Form Elements from "Historic" Architectural Facade Employing Inductive Logic Programming. T. Washio et al. (Eds.): *JSAI 2005 Workshops, LNAI 4012*, 190–200, 2006. Springer-Verlag, Berlin-Heidelberg.

[3] Yong Liu, Congfu Xu, Qiong Zhang, and Yunhe Pan. The smart architect: Scalable ontology-based modeling of ancient Chinese architectures. *IEEE Intelligent Systems*, 23:49–56, 2008.

[4] Yong Liu, Yunliang Jiang, and Lican Huang. Modeling Complex Architectures Based on Granular Computing on Ontology. *IEEE Transactions on Fuzzy Systems,* vol. 18, no. 3, 585–598, June 2010

[5] Paul Frankl. *Gothic Architecture.* Revised by Paul Crossley. Yale University Press: New Haven, 2000.

[6] Charles Hollingsworth, Stefaan Van Liefferinge, Rebecca A. Smith, Michael A. Covington, and Walter D. Potter. The ARC Project: Creating logical models of Gothic cathedrals using natural language processing. *Proceedings of ACL 2011 Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities* (forthcoming)

[7] Patrick Blackburn and Johan Bos. *Representation and inference for natural language: A first course in computational semantics*. CSLI Publications, Stanford, CA, USA, 1999.

[8] Michael A. Covington, Donald Nute, Nora Schmitz, and David Goodman. *From English to Prolog via discourse representation theory.* Technical report, Advanced Computational Methods Center, The University of Georgia, 1988. http://www.ai.uga.edu/ftplib/ai-reports/ai010024.pdf (available as of May 2011).

[9] Thoralf Skolem. Über die mathematische Logik (Nach einem Vortrag gehalten im Norwegischen Mathematischen Verien am 22. Oktober 1928). In *Selected Works in Logic.* Jens Erik Fenstad, ed. Universitetsforlaget, Oslo - Bergen - Tromsö, 1970, 189–206.

[10] Jeffrey Cua, Ruli Manurung, Ethel Ong, and Adam Pease. Representing Story plans in SUMO. In *Proceedings of the NAACL HLT 2010 Second Workshop on Computational Approaches to Linguistic Creativity.* Association for Computational Linguistics, Los Angeles, California, June 2010, 40–48.

[11] Grigoris Antoniou. *Nonmonotonic Reasoning.* The MIT Press, Cambridge, MA, USA, 1997.

[12] Raymond Reiter. Nonmonotonic Reasoning. *Annual Review of Computer Science,* 1987.2: 147–186.

[13] Donald Nute. Defeasible Logic. In *Proceedings of the Applications of Prolog 14th International Conference on Web Knowledge Management and Decision Support* (INAP'01). Oskar Bartenstein, Ulrich Geske, Markus Hannebauer, and Osamu Yoshie, eds. Springer-Verlag, Berlin-Heidelberg, 151–169.

An Automated Derivation of Church's P_2 Sentential Calculus from Łukasiewicz's *CN*

Jack K. Horner
P. O. Box 266
Los Alamos, New Mexico 87544 USA

Abstract

Two logics are implicationally equivalent if the axioms and inference rules of each imply the axioms of the other. Characterizing equivalences of various formulations of the sentential calculi is foundational to the study of logic. Using an automated deduction system, I show that Church's P_2 sentential calculus can be derived from Łukasiewicz's CN. To prove each of the axioms of P_2, the deduction system automatically develops a single set lemmas and propositions -- an elementary "core" theory. Yhe proof appears to be novel.

Keywords: propositional logic, automated deduction, sentential calculus

1.0 Introduction

Characterizing equivalences of various formulations of the sentential calculi is foundational to the study of logic ([1],[3]-[7],[9]-[10],[12]-[15]).

"CN", the formulation of the sentential calculus in [1], is among the most austere: its vocabulary contains only two logical connectives (C, and N) and sentence variables (*p, q, r,* ...). It has two inference rules (condensed detachment and substitution), and three axioms.

In CN, any expression of the form *Cxy* or *Nz*, where *x, y,* and *z* are sentences, is a sentence. *Cpq* is interpreted as "sentence *p* implies sentence *q*"; *Np* is interpreted as "not-*p*". N and C are right-associative; N has higher associative precedence than C. For example,

CCqrCpNr

translates to the more common "arrow-and-parenthesis" notation as

$$(q \rightarrow r) \rightarrow (p \rightarrow \sim r)$$

where "$\rightarrow$" designates "implies" and "~" designates "not".

The axioms of CN in [1] are:

CN1. *CCpqCCqrCpr*
CN2. *CCNppp*
CN3. *CpCNpq*

Cast in CN notation, the axioms of the Church's P_2 sentential calculus (P2, [12]) are

P2-202. *CpCqp*
P2-203. *CCsCpqCCspCsq*
P2-204. *CCNpNqCqp*

The main result of this paper is that [1] implies [12].

2.0 Method

To show that [1] implies [12], the *prover9* ([2]) script shown in Figures 1 was executed

under on a Dell Inspiron 545 with an Intel Core2 Quad CPU Q8200 @ 2.33 GHz and 8.00 GB RAM, running under the *Windows Vista Home Premium (SP2)/Cygwin* operating environment.

```
set(hyper_resolution).

formulas(usable).
P ( i(i(u, i(y, i(x,z))), i(i(x,u), i(y, i(x,z)))) )  # label("CN34").
P( i(i(x,y), i(i(y,z), i(x,z))) )      # label("AxCN1").
P( i(i(-x,x), x) )                     # label("AxCN2").
P( i(x, i(-x,y)) )                     # label("AxCN3").
end_of_list.

formulas(sos).
-P(i(x,y)) | -P(x) | P(y)          # label("InfConDet").
end_of_list.

formulas(goals).
P ( i(x, i(y,x)) )                          # label("P2-202").
P ( i(i(z, i(x,y)), i(i(z,x), i(z,y))) )  # label("P2-203").
P ( i(i(-x,-y), i(y,x)) )                   # label("P2-204").
end_of_list.
```

Figure 1. The *prover9* script (in Horn clause ([11]) form) used to show that CN implies P2. The implementation of condensed detachment is the formula in the "sos" list; substitution is derived from *prover9*'s hyperresolution rule (introduced in the "set" command at the top of the script). CN34, a theorem of [1], has been added to the axioms of [1] to facilitate the derivation. Details of *prover9*'s syntax and semantics can be found in [2].

3.0 Results

Figure 2 shows that CN implies P2.

```
============================== PROOF =================================

% Proof 1 at 2.25 (+ 0.09) seconds: "P2-202".
% Length of proof is 41.
% Level of proof is 20.
% Maximum clause weight is 16.
% Given clauses 497.

1 P(i(x,i(y,x))) # label("P2-202") # label(non_clause) # label(goal).
[goal].
4 P(i(i(x,i(y,i(z,u))),i(i(z,x),i(y,i(z,u))))) # label("CN34").
[assumption].
5 P(i(i(x,y),i(i(y,z),i(x,z)))) # label("AxCN1").  [assumption].
6 P(i(i(-x,x),x)) # label("AxCN2").  [assumption].
```

```
7 P(i(x,i(-x,y))) # label("AxCN3").  [assumption].
8 -P(i(x,y)) | -P(x) | P(y) # label("InfConDet").  [assumption].
9 -P(i(c1,i(c2,c1))) # label("P2-202") # answer("P2-202").  [deny(1)].
16 P(i(i(i(-x,y),z),i(x,z))).  [hyper(8,a,5,a,b,7,a)].
17 P(i(i(x,y),i(i(-x,x),y))).  [hyper(8,a,5,a,b,6,a)].
18 P(i(i(i(i(x,y),i(z,y)),u),i(i(z,x),u))).  [hyper(8,a,5,a,b,5,a)].
20 P(i(i(x,y),i(-y,i(x,z)))).  [hyper(8,a,4,a,b,7,a)].
34 P(i(x,x)).  [hyper(8,a,16,a,b,6,a)].
37 P(i(-i(x,x),y)).  [hyper(8,a,7,a,b,34,a)].
40 P(i(i(x,y),i(-i(z,z),y))).  [hyper(8,a,5,a,b,37,a)].
90 P(i(i(i(-x,i(y,z)),u),i(i(y,x),u))).  [hyper(8,a,5,a,b,20,a)].
95 P(i(-x,i(x,y))).  [hyper(8,a,20,a,b,34,a)].
111 P(i(i(i(x,y),z),i(-x,z))).  [hyper(8,a,5,a,b,95,a)].
140 P(i(i(x,i(y,z)),i(i(u,y),i(x,i(u,z))))).  [hyper(8,a,18,a,b,18,a)].
229 P(i(x,i(-i(y,y),z))).  [hyper(8,a,16,a,b,40,a)].
236 P(i(i(i(-i(x,x),y),z),i(u,z))).  [hyper(8,a,5,a,b,229,a)].
337 P(i(--x,x)).  [hyper(8,a,111,a,b,6,a)].
344 P(i(i(x,y),i(--x,y))).  [hyper(8,a,5,a,b,337,a)].
401 P(i(x,i(---x,y))).  [hyper(8,a,16,a,b,344,a)].
3638 P(i(x,i(y,y))).  [hyper(8,a,236,a,b,6,a)].
3657 P(i(i(i(x,x),y),i(z,y))).  [hyper(8,a,5,a,b,3638,a)].
3715 P(i(i(x,y),i(z,i(x,y)))).  [hyper(8,a,18,a,b,3657,a)].
3733 P(i(x,i(i(-y,y),y))).  [hyper(8,a,3657,a,b,17,a)].
3877 P(i(i(x,i(x,y)),i(z,i(x,y)))).  [hyper(8,a,4,a,b,3715,a)].
5579 P(i(x,i(i(y,i(y,z)),i(y,z)))).  [hyper(8,a,3877,a,b,3877,a)].
6330 P(i(i(x,i(x,y)),i(x,y))).  [hyper(8,a,5579,a,b,5579,a)].
6378 P(i(i(i(x,y),x),i(i(x,y),y))).  [hyper(8,a,18,a,b,6330,a)].
6747 P(i(i(i(i(-x,x),x),y),y)).  [hyper(8,a,6378,a,b,3733,a)].
6791 P(i(i(x,i(-y,y)),i(x,y))).  [hyper(8,a,18,a,b,6747,a)].
6939 P(i(x,--x)).  [hyper(8,a,6791,a,b,401,a)].
6954 P(i(x,i(y,--y))).  [hyper(8,a,3715,a,b,6939,a)].
7340 P(i(i(-x,y),i(-y,x))).  [hyper(8,a,90,a,b,6791,a)].
8116 P(i(i(i(x,--x),y),y)).  [hyper(8,a,6378,a,b,6954,a)].
9627 P(i(x,i(-y,x))).  [hyper(8,a,16,a,b,7340,a)].
9700 P(i(i(x,-y),i(z,i(x,z)))).  [hyper(8,a,140,a,b,9627,a)].
18819 P(i(x,i(y,x))).  [hyper(8,a,8116,a,b,9700,a)].
18820 $F # answer("P2-202").  [resolve(18819,a,9,a)].

============================== end of proof ==========================

============================== PROOF =================================

% Proof 2 at 7.35 (+ 0.45) seconds: "P2-204".
% Length of proof is 49.
% Level of proof is 26.
% Maximum clause weight is 16.
% Given clauses 1282.

3 P(i(i(-x,-y),i(y,x))) # label("P2-204") # label(non_clause) #
label(goal).  [goal].
4 P(i(i(x,i(y,i(z,u))),i(i(z,x),i(y,i(z,u))))) # label("CN34").
[assumption].
5 P(i(i(x,y),i(i(y,z),i(x,z)))) # label("AxCN1").  [assumption].
6 P(i(i(-x,x),x)) # label("AxCN2").  [assumption].
7 P(i(x,i(-x,y))) # label("AxCN3").  [assumption].
8 -P(i(x,y)) | -P(x) | P(y) # label("InfConDet").  [assumption].
```

```
11 -P(i(i(-c6,-c7),i(c7,c6))) # label("P2-204") # answer("P2-204").
[deny(3)].
16 P(i(i(i(-x,y),z),i(x,z))).  [hyper(8,a,5,a,b,7,a)].
17 P(i(i(x,y),i(i(-x,x),y))).  [hyper(8,a,5,a,b,6,a)].
18 P(i(i(i(i(x,y),i(z,y)),u),i(i(z,x),u))).  [hyper(8,a,5,a,b,5,a)].
20 P(i(i(x,y),i(-y,i(x,z)))).  [hyper(8,a,4,a,b,7,a)].
34 P(i(x,x)).  [hyper(8,a,16,a,b,6,a)].
37 P(i(-i(x,x),y)).  [hyper(8,a,7,a,b,34,a)].
40 P(i(i(x,y),i(-i(z,z),y))).  [hyper(8,a,5,a,b,37,a)].
90 P(i(i(i(-x,i(y,z)),u),i(i(y,x),u))).  [hyper(8,a,5,a,b,20,a)].
95 P(i(-x,i(x,y))).  [hyper(8,a,20,a,b,34,a)].
111 P(i(i(i(x,y),z),i(-x,z))).  [hyper(8,a,5,a,b,95,a)].
140 P(i(i(x,i(y,z)),i(i(u,y),i(x,i(u,z))))).  [hyper(8,a,18,a,b,18,a)].
229 P(i(x,i(-i(y,y),z))).  [hyper(8,a,16,a,b,40,a)].
236 P(i(i(i(-i(x,x),y),z),i(u,z))).  [hyper(8,a,5,a,b,229,a)].
337 P(i(--x,x)).  [hyper(8,a,111,a,b,6,a)].
344 P(i(i(x,y),i(--x,y))).  [hyper(8,a,5,a,b,337,a)].
401 P(i(x,i(---x,y))).  [hyper(8,a,16,a,b,344,a)].
3638 P(i(x,i(y,y))).  [hyper(8,a,236,a,b,6,a)].
3657 P(i(i(i(x,x),y),i(z,y))).  [hyper(8,a,5,a,b,3638,a)].
3715 P(i(i(x,y),i(z,i(x,y)))).  [hyper(8,a,18,a,b,3657,a)].
3733 P(i(x,i(i(-y,y),y))).  [hyper(8,a,3657,a,b,17,a)].
3877 P(i(i(x,i(x,y)),i(z,i(x,y)))).  [hyper(8,a,4,a,b,3715,a)].
5579 P(i(x,i(i(y,i(y,z)),i(y,z)))).  [hyper(8,a,3877,a,b,3877,a)].
6330 P(i(i(x,i(x,y)),i(x,y))).  [hyper(8,a,5579,a,b,5579,a)].
6378 P(i(i(i(x,y),x),i(i(x,y),y))).  [hyper(8,a,18,a,b,6330,a)].
6747 P(i(i(i(i(-x,x),x),y),y)).  [hyper(8,a,6378,a,b,3733,a)].
6791 P(i(i(x,i(-y,y)),i(x,y))).  [hyper(8,a,18,a,b,6747,a)].
6906 P(i(i(-x,y),i(i(y,x),x))).  [hyper(8,a,18,a,b,6791,a)].
6939 P(i(x,--x)).  [hyper(8,a,6791,a,b,401,a)].
6954 P(i(x,i(y,--y))).  [hyper(8,a,3715,a,b,6939,a)].
6980 P(i(i(--x,y),i(x,y))).  [hyper(8,a,5,a,b,6939,a)].
7340 P(i(i(-x,y),i(-y,x))).  [hyper(8,a,90,a,b,6791,a)].
8116 P(i(i(i(x,--x),y),y)).  [hyper(8,a,6378,a,b,6954,a)].
9627 P(i(x,i(-y,x))).  [hyper(8,a,16,a,b,7340,a)].
9700 P(i(i(x,-y),i(z,i(x,z)))).  [hyper(8,a,140,a,b,9627,a)].
18819 P(i(x,i(y,x))).  [hyper(8,a,8116,a,b,9700,a)].
18900 P(i(i(i(x,y),z),i(y,z))).  [hyper(8,a,5,a,b,18819,a)].
20411 P(i(x,i(i(x,y),y))).  [hyper(8,a,18900,a,b,6906,a)].
20878 P(i(i(x,i(y,z)),i(y,i(x,z)))).  [hyper(8,a,140,a,b,20411,a)].
43771 P(i(-x,i(i(-y,x),y))).  [hyper(8,a,20878,a,b,7340,a)].
44691 P(i(x,i(i(-y,-x),y))).  [hyper(8,a,6980,a,b,43771,a)].
45362 P(i(i(-x,-y),i(y,x))).  [hyper(8,a,20878,a,b,44691,a)].
45363 $F # answer("P2-204").  [resolve(45362,a,11,a)].

============================== end of proof ==========================


============================== PROOF =================================

% Proof 3 at 10.53 (+ 0.80) seconds: "P2-203".
% Length of proof is 47.
% Level of proof is 25.
% Maximum clause weight is 16.
% Given clauses 1658.
```

```
2 P(i(i(z,i(x,y)),i(i(z,x),i(z,y)))) # label("P2-203") #
label(non_clause) # label(goal).  [goal].
4 P(i(i(x,i(y,i(z,u))),i(i(z,x),i(y,i(z,u))))) # label("CN34").
[assumption].
5 P(i(i(x,y),i(i(y,z),i(x,z)))) # label("AxCN1").  [assumption].
6 P(i(i(-x,x),x)) # label("AxCN2").  [assumption].
7 P(i(x,i(-x,y))) # label("AxCN3").  [assumption].
8 -P(i(x,y)) | -P(x) | P(y) # label("InfConDet").  [assumption].
10 -P(i(i(c3,i(c4,c5)),i(i(c3,c4),i(c3,c5)))) # label("P2-203") #
answer("P2-203").  [deny(2)].
16 P(i(i(i(-x,y),z),i(x,z))).  [hyper(8,a,5,a,b,7,a)].
17 P(i(i(x,y),i(i(-x,x),y))).  [hyper(8,a,5,a,b,6,a)].
18 P(i(i(i(i(x,y),i(z,y)),u),i(i(z,x),u))).  [hyper(8,a,5,a,b,5,a)].
20 P(i(i(x,y),i(-y,i(x,z)))).  [hyper(8,a,4,a,b,7,a)].
34 P(i(x,x)).  [hyper(8,a,16,a,b,6,a)].
37 P(i(-i(x,x),y)).  [hyper(8,a,7,a,b,34,a)].
40 P(i(i(x,y),i(-i(z,z),y))).  [hyper(8,a,5,a,b,37,a)].
90 P(i(i(i(-x,i(y,z)),u),i(i(y,x),u))).  [hyper(8,a,5,a,b,20,a)].
95 P(i(-x,i(x,y))).  [hyper(8,a,20,a,b,34,a)].
111 P(i(i(i(x,y),z),i(-x,z))).  [hyper(8,a,5,a,b,95,a)].
140 P(i(i(x,i(y,z)),i(i(u,y),i(x,i(u,z))))).  [hyper(8,a,18,a,b,18,a)].
229 P(i(x,i(-i(y,y),z))).  [hyper(8,a,16,a,b,40,a)].
236 P(i(i(i(-i(x,x),y),z),i(u,z))).  [hyper(8,a,5,a,b,229,a)].
337 P(i(--x,x)).  [hyper(8,a,111,a,b,6,a)].
344 P(i(i(x,y),i(--x,y))).  [hyper(8,a,5,a,b,337,a)].
401 P(i(x,i(---x,y))).  [hyper(8,a,16,a,b,344,a)].
3638 P(i(x,i(y,y))).  [hyper(8,a,236,a,b,6,a)].
3657 P(i(i(i(x,x),y),i(z,y))).  [hyper(8,a,5,a,b,3638,a)].
3715 P(i(i(x,y),i(z,i(x,y)))).  [hyper(8,a,18,a,b,3657,a)].
3733 P(i(x,i(i(-y,y),y))).  [hyper(8,a,3657,a,b,17,a)].
3877 P(i(i(x,i(x,y)),i(z,i(x,y)))).  [hyper(8,a,4,a,b,3715,a)].
5579 P(i(x,i(i(y,i(y,z)),i(y,z)))).  [hyper(8,a,3877,a,b,3877,a)].
6330 P(i(i(x,i(x,y)),i(x,y))).  [hyper(8,a,5579,a,b,5579,a)].
6378 P(i(i(i(x,y),x),i(i(x,y),y))).  [hyper(8,a,18,a,b,6330,a)].
6747 P(i(i(i(i(-x,x),x),y),y)).  [hyper(8,a,6378,a,b,3733,a)].
6791 P(i(i(x,i(-y,y)),i(x,y))).  [hyper(8,a,18,a,b,6747,a)].
6906 P(i(i(-x,y),i(i(y,x),x))).  [hyper(8,a,18,a,b,6791,a)].
6939 P(i(x,--x)).  [hyper(8,a,6791,a,b,401,a)].
6954 P(i(x,i(y,--y))).  [hyper(8,a,3715,a,b,6939,a)].
7340 P(i(i(-x,y),i(-y,x))).  [hyper(8,a,90,a,b,6791,a)].
8116 P(i(i(i(x,--x),y),y)).  [hyper(8,a,6378,a,b,6954,a)].
9627 P(i(x,i(-y,x))).  [hyper(8,a,16,a,b,7340,a)].
9700 P(i(i(x,-y),i(z,i(x,z)))).  [hyper(8,a,140,a,b,9627,a)].
18819 P(i(x,i(y,x))).  [hyper(8,a,8116,a,b,9700,a)].
18900 P(i(i(i(x,y),z),i(y,z))).  [hyper(8,a,5,a,b,18819,a)].
20411 P(i(x,i(i(x,y),y))).  [hyper(8,a,18900,a,b,6906,a)].
20878 P(i(i(x,i(y,z)),i(y,i(x,z)))).  [hyper(8,a,140,a,b,20411,a)].
43854 P(i(i(x,y),i(i(z,x),i(z,y)))).  [hyper(8,a,20878,a,b,5,a)].
54576 P(i(i(x,i(y,z)),i(i(x,y),i(x,z)))).  [hyper(8,a,4,a,b,43854,a)].
54577 $F # answer("P2-203").  [resolve(54576,a,10,a)].

============================== end of proof ==========================
```

Figure 2. Summary of a *prover9* ([2]) proof showing that CN ([1]) implies P2 ([12]).

The total time to complete the proofs shown in Figure 2 was ~21 seconds on the platform described in Section 2.0.

4.0 Conclusions and discussion

Section 3 demonstrates that CN implies P_2. A companion paper ([16]) proves P2 implies CN. The proof appears to be novel.

There are some interesting relations among the proofs in Section 3. To prove each of the axioms of P_2, the deduction system used first developed a shared set lemmas and propositions from CN -- an elementary "core" theory ending at Line 6954 in each proof. In addition, the proof of P2-202 through Line 18819 is a subproof of the proof of P2-204; thus, that subproof may be regarded as a set of lemmas for the proof of P2-204. Similarly, the proof of P2-204 through Line 20878 is a subproof of the proof of P2-203; thus, that subproof may be regarded as a set of lemmas for the proof of P2-203.

5.0 References

[1] Łukasiewicz J. *Elements of Mathematical Logic*. Second Edition (1958). Trans. by Wojtasiewicz O. Pergamon Press. 1963.

[2] McCune WW. *prover9 and mace4.* http://www.cs.unm.edu/~mccune/prover9/. 2009.

[3] Aristotle. *Prior Analytics*. Trans. by A. J. Jenkinson. In Aristotle. *The Basic Works of Aristotle.* Ed. by R. McKeon. Random House. 1941. pp. 62-107.

[4] Aristotle. *Posterior Analytics.* Trans. by G. R. G. Mure. In Aristotle. *The Basic Works of Aristotle.* Ed. by R. McKeon. Random House. 1941. pp. 108-186.

[5] Tarski A. *Introduction to Logic*. Trans. by O. Helmer. Dover. 1941.

[6] Hempel C. Studies in the logic of explanation. In Hempel C. *Aspects of Scientific Explanation and Other Essays in the Philosophy of Science.* Free Press. 1965. pp. 245-290.

[7] Quine WVO. *Philosophy of Logic*. Second Edition. Harvard. 1986.

[8] Chang CC and Keisler HJ. *Model Theory*. North-Holland. 1990.

[9] Russell B and Whitehead AN. *Principia Mathematica*. Volume I (1910). Merchant Books. 2009.

[10] Frege G. Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens. Halle. 1879. Translated in van Heijenoort J. Begriffsschrift, a formula language, modeled upon that of arithmetic, for pure thought. *From Frege to Gödel: A Source Book in Mathematical Logic, 1879-1931*. Harvard. 1967. pp. 3-82.

[11] Horn A. On sentences which are true of direct unions of algebras. *Journal of Symbolic Logic* 16 (1951), 14–21.

[12] Church A. *Introduction to Mathematical Logic*. Volume I. Princeton. 1956.

[13] Birkhoff G and von Neumann J. The logic of quantum mechanics. *Annals of Mathematics* 37 (1936), 823-843.

[14] Kant I. *Kant's Introduction to Logic* (1800). Trans. by Abbott TK. Greenwood Press. 1963.

[15] Cohen MR and Nagel E. *An Introduction to Logic and Scientific Method.* Harcourt, Brace, and Company. 1934.

[16] Horner JK. An automated derivation of Łukasiewicz's CN sentential calculus from Church's P_2. *Proceedings of the 2011 International Conference on Artificial Intelligence.* CSREA Press. Forthcoming.

Optimizing In-Memory Joins for a Distributed Deductive Database Running in a Multi-Core Environment

Martin Maskarinec[1] and Kathleen Neumann[2]
[1]School of Computer Sciences, Western Illinois University, Macomb, Illinois, USA
[2]School of Computer Sciences, Western Illinois University, Macomb, Illinois, USA

Abstract
Distributed deductive databases can expand their efficiency within a multi-core environment by optimizing the in-memory join operations by minimizing memory transfers. The paper presents the algorithm and a calculation of the savings for this type of optimization.

Keywords: Intelligent Database, Deductive Database, Join Algorithm

1. Introduction

A deductive database uses logic programming as the deductive component of a relational database and allows for extensions that make that database more complex than a traditional relational database [1]. To aid in this process, we view our base-relations and rule predicates in a graph known as the *Extended Predicate Connection Graph* (EPCG). Figure 1 represents an example EPCG. The links in the graph represent two-way communication paths. Information can travel between nodes that have a link/path. The EPCG is partitioned and distributed to other machines that act as a deductive database server or even to clusters of machines or multi-core machines to gain as much efficiency as possible through simultaneous computation. Each partition may contain one or more base-relations as well as the set of rules or predicates. By partitioning the EPCG and distributing the partitions, we are able to process user queries in a distributed fashion, and more than one query can be active within the system at any given point in time. An algorithm has been developed to deliver partial results as they are being found [5]. This means that all of necessary components within a partition, including the sorts [4] and the joins, can be done in parallel on different cores.

One of the *Extended* features is the ability to optimize the in-memory join operations for a distributed deductive database running in a multi-core environment. The optimization is based on the ability to minimize the number of in-memory transfers. This paper presents the algorithm and a calculation of the savings for this type of optimization.

2. Overview

Suppose I have a predicate definition as follows:

$\mathbf{P_1}$ (U_1, U_2, U_3, U_4, U_5, U_6, U_7, U_8, U_9, U_{10}, U_{11}, U_{12}, U_{13}, U_{14},) :- $\mathbf{B_1}$ (A, U_1, U_2, U_3), $\mathbf{B_2}$ (A, B, U_4, U_5), $\mathbf{B_3}$ (B, C, U_5, U_6, U_7, U_8,), $\mathbf{B_3}$ (C, D, U_5, U_6, U_7, U_8,), $\mathbf{B_4}$ (D, E, U_9, U_{10},), $\mathbf{B_5}$ (E, U_{11}, U_{12}, U_{13}, U_{14},)

Normal query optimization would allow me to remove unneeded attributes before processing, saving time and space. In P_1 however, all attributes are needed, either in the answer or in the join, so no intermediate projection could be used. However, I can use B_1 (A, Row_{B1}), B_2 (A, B Row_{B2}), B_3 (B, C, Row_{B3}), B_4 (C, D, Row_{B4}), B_5 (D, Row_{B5}). This substantially reduces the size of the intermediate results. After finishing this new version of the query, I would have

Result (A, B, C, D, E, Row_{B1}, Row_{B2}, Row_{B3}, Row_{B4})

For each row in Result, I would now need to look up Row_{B1}…Row_{B5} to get U1…U14

3. Generating the Result

3.1 Normal Way

In normal processing, I might use a sort-merge-join to produce the result of the query. This would follow the following steps:

1) Sort each B_i on the join-key with B_{i-1} (B_1 sorted on join-key with B_2)
2) For each record in B_1, look it up in B_2 then B_3, B_4, …… B_n
3) Return Results

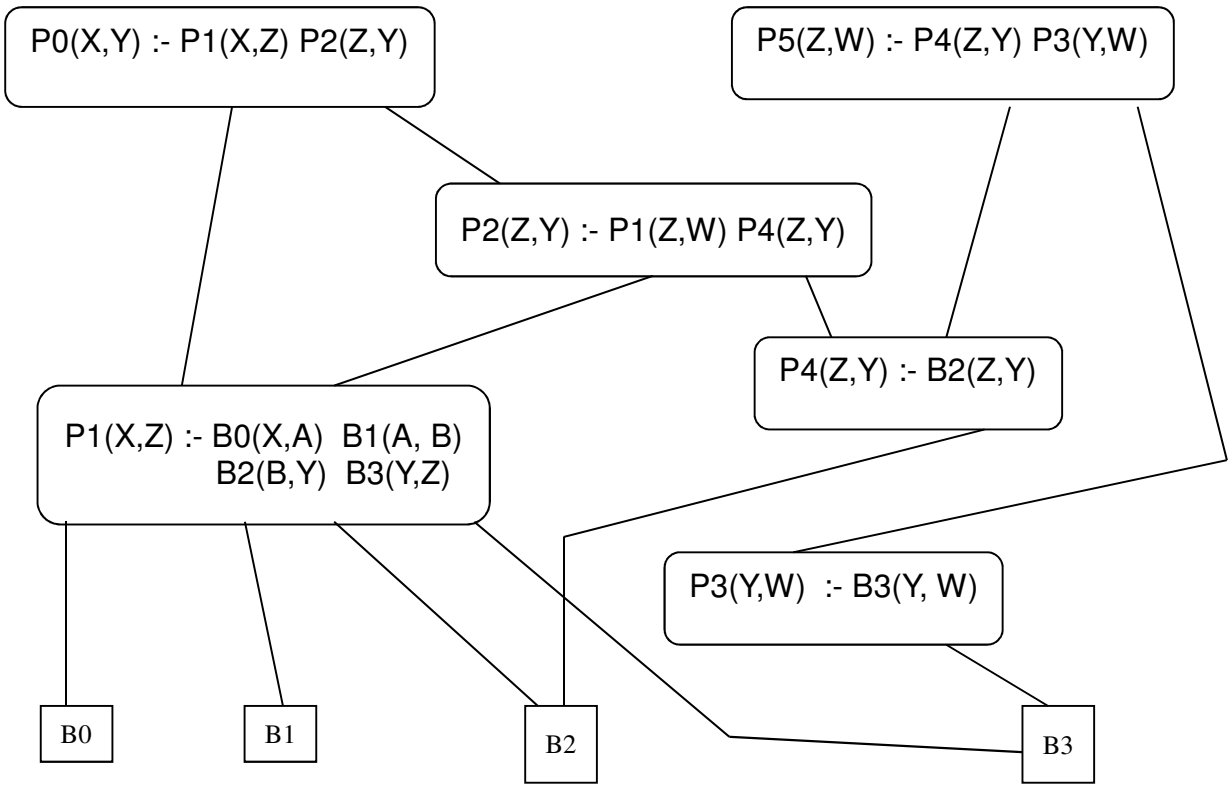


Figure 1. An Extended Predicate Connection Graph.

3.2 New Way

In a multi-core environment, I may be able to save memory transfers using the following steps:

1) Sort each B_i on the join-key, but store only join key and the row number
2) For each record in B_1, look it up in B_2 then B_3, B_4, …… B_n
3) Join Result in Step 2 with the original data to retrieve the required columns
4) Return Results

4. Cost Analysis

4.1 Preparation Savings

Performing the sorts in step 1) with only the join key and the row number saves space (and therefore memory access time as lesser amounts of data may be transferred faster). In this section we determine how much space may be saved.

First, let us analyze one pass of the merge sort. For each predicate, one pass saves

$$\textit{One Pass Savings}_{P_i} = (\textit{Number Of Bytes Saved/Record}_{P_i}) * (r_{P_i})$$

Where P_i refers to the i^{th} predicate and r_{P_i} refers to the number of records in this predicate. The number of bytes saved per record may be calculated as

$$\textit{Number Of Bytes Saved/Record}_{P_i} = (\textit{Total bytes/record}_{P_i}) - \textit{Join Attribute Size}_{P_i} - \textit{Row Number Size}$$

This equation is for one pass of the merge sort; since the number of passes of merge sort is $\log_2(r_{P_i})$, we may calculate the total preparation savings as

$$\textit{Preparation Savings}_{P_i} = [\log_2(r_{P_i}) - 1] * (\textit{One Pass Savings}_{P_i})$$

Here, we need to subtract 1 as the first pass would need to be done on the full data set. Finally, we can calculate the

total preparation savings by summing up the individual predicate preparation savings as in the following equation:

$$Total\ Preparation\ Savings = \sum_{i=1}^{n} Preparation\ Savings_{Pi}$$

4.2 Join Savings

When performing multiple joins, there is no way to have both parts being joined sorted on the same attribute. For example, in our sample predicate given in Section 2, we can sort B1 and B2 on attribute A, but this result cannot be generated as sorted on attribute B for the next join with B3. Thus, the B1-B2 result must be treated as unsorted on attribute B and thus each row must be looked up in the sorted list B3. Once again, we can save bytes by having only the row numbers and the join attributes in these intermediate results. The number of byte saved doing the join in this way is given by

$$Join\ Savings = Join\ Real - Join\ Project$$

$$Join\ Real = [\sum_{i=1}^{n-1} |r_{Ji}| * r_{Ji} * |r_{P(i+1)}| * \log_2(r_{P(i+1)}) + |r_{J(i+1)}| * r_{J(i+1)}]$$

$$Join\ Project = [\sum_{i=1}^{n-1} |rp_{Ji}| * r_{Ji} * |rp_{P(i+1)}| * \log_2(r_{P(i+1)}) + |rp_{J(i+1)}| * r_{J(i+1)}]$$

where

r_{Pi} refers to the number of records in Predicate i
r_{Ji} refers to the number of records in join result i
$|r_{Pi}|$ refers to the number of bytes in Prediate i
$|r_{Ji}|$ refers to the number of bytes in join result i
$|rp_{Pi}| = |r_{Pi}|$ – number of bytes saved per record in predicate i
$|rp_{Ji}| = |r_{Ji}|$ – number of bytes saved per record in join result i
and, for convenience, we let $r_{j1} = r_{p1}$, and $rp_{j1} = rp_{p1}$.

4.3 Additional End Costs

The downside of this technique is that it does require extra time at the end to retrieve the actual attributes from just the row numbers that come out of the sort/join result. We do this, following these steps:

[Starting with the join keys and $RecordNumber_{Pi}$]:

1) Identify columns needed from P_i which are not in the join-key set
2) Any P_i which has no additional attributed needed may be discarded
3) For each remaining P_i there will be one disk access for every record in the result. Thus the cost of these joins would be:

$$r_{result} * [\sum_{j=1}^{m} |r_{Pj}|]$$

where m is the number of predicates having matches.

4.4 Final Analysis

Finally, we now may determine if this technique will be beneficial to us. To do this, we calculate the overall savings as

$$Overall\ Savings = Total\ Prep\ Savings + Join\ Savings - Total\ Additional\ Cost$$

If this equation is positive, then we will save memory accesses (and therefore time) by performing this algorithm. If it is negative, we should just process the join in the normal way as the added additional cost will outweigh the savings.

5. Results

There are many different values presented in the previous section, so a spreadsheet was developed to determine the total number of bytes which may be saved using this technique. The initial configuration was to have 1 million records in the first predicate and 1,000 records in predicates 2 through 5 (this would be equivalent to a relatively small relational data warehouse). With all predicates being 50 bytes long, join columns being 10 bytes long, 4 byte numbers, and 1 expected match for each join, this technique would require approximately 192 billion fewer bytes to be transferred through the memory bus.

Ratcheting up the number of rows in predicates 2 through 5 causes the savings to grow to 211 billion bytes; ratcheting up the expected number of matches causes the savings to grow to 932 billion bytes. Clearly, the number of expected matches has a dramatic effect on the savings. This is not surprising as the number of records that need to be saved doubles with each join in this example.

On the negative side, decreasing the number of bytes of non-join attributes clearly would decrease the savings. For example, reducing the number of bytes of each predicate to 30 bytes total results in a savings of just 44 billion bytes and reducing the size to 25 bytes total results in a savings of only 19 billion bytes.

6. Conclusions

We have presented a method of optimizing memory bus usage in a multi-core environment for the types of joins we generally see in deductive databases. This method carries along only a record number for intermediate results, and then joins these back to the original data after all joins have been done. This can result in significant savings when a large number of attributes (or a smaller number of large attributes) would normally need to be carried through multiple joins. This method would not perform well on smaller results, or ones which need only the join attributes anyway. Finally, we are assuming the only cost is memory transfer – no CPU processing time was considered. However, in a multi core environment with a sufficient number of cores, this assumption should be valid for an I/O intensive application like a deductive database.

7. References

[1] Ramakrishnan, Raghu and Johannes Gehrke. *Database Management Systems Third Edition* McGraw-Hill. 2003.

[2] Neumann, Kathleen and Lawrence Henschen. "Partitioning Algorithms for a Distributed Deductive Database." *ACM Computer Science Conference Proceedings.* 1994.

[3] Maskarinec, Martin and Kathleen Neumann. "Heterogeneous Cost Functions and a Partitioning Algorithm for a Distributed Deductive Database" *Proceeding of the 2003 International Conference on Artificial Intelligence.*

[4] Maskarinec, Martin and Kathleen Neumann. "Optimized Sorting Functions for a Distributed Deductive Database" *Proceeding of the 2008 International Conference on Artificial Intelligence.*

[5] Maskarinec, Martin and Kathleen Neumann. **"**Comparison of Two Parallel Processing Methods for a Distributed Deductive Database**"** *Proceeding of the 2010 International Conference on Artificial Intelligence.*

An Automated Derivation of Frege's Sentential Calculus from Łukasiewicz's *CN*

Jack K. Horner
P. O. Box 266
Los Alamos, New Mexico 87544 USA

Abstract

Two logics are implicationally equivalent if the axioms and inference rules of each imply the axioms of the other. Characterizing the inferential equivalences of various formulations of the sentential calculi is foundational to the study of logic. Using an automated deduction system, I show that the sentential calculus of Frege's Begriffsschrift, *the first sentential calculus, can be derived from Łukasiewicz's CN; the proof appears to be novel.*

Keywords: propositional logic, automated deduction, sentential calculus

1.0 Introduction

Two logics are implicationally equivalent if the axioms and inference rules of each imply the axioms of the other. Characterizing the inferential equivalences of various formulations of the sentential calculi is thus foundational to the study of logic ([1],[3]-[7],[9]-[10],[12]-[15]),.

"CN", the formulation of the sentential calculus in [1], is among the most austere: its vocabulary contains only two logical connectives (C, and N) and sentence variables (*p, q, r,* ...). It has two inference rules (condensed detachment and substitution), and three axioms.

In CN, any expression of the form *Cxy* or *Nz*, where *x, y,* and *z* are sentences, is a sentence. *Cpq* is interpreted as "sentence *p* implies sentence *q*"; *Np* is interpreted as "not-*p*". C and N are right-associative; N has higher associative precedence than C. For example,

CCqrCpNr

translates to the more common "arrow-and-parenthesis" notation as

$$(q \rightarrow r) \rightarrow (p \rightarrow \sim r)$$

where "$\rightarrow$" designates "implies" and "~" designates "not".

The axioms of CN in [1] are:

CN1. *CCpqCCqrCpr*
CN2. *CCNppp*
CN3. *CpCNpq*

Cast in CN notation, the axioms of Frege's *Begriffsschrift* (BG, [10]), the first sentential calculus, are

CN18. *CqCpq*
CN21. *CCpCqrCqCpr*
CN35. *CCpCqrCCpqCpr*
CN39. *CNNpp*
CN40. *CpNNp*
CN46. *CCpqCNqNp*

The main result of this paper is that [1] implies [10].

2.0 Method

To show that [1] implies [10], the *prover9* ([2]) script shown in Figures 1 was executed under on a Dell Inspiron 545 with an Intel Core2 Quad CPU Q8200 @ 2.33 GHz and 8.00 GB RAM, running under the *Windows Vista Home Premium (SP2)/Cygwin* operating environment.

```
set(hyper_resolution).

formulas(usable).
P ( i(i(u, i(y, i(x,z))), i(i(x,u), i(y, i(x,z)))) )  # label("CN34").
P( i(i(x,y), i(i(y,z), i(x,z))) )      # label("AxCN1").
P( i(i(-x,x), x) )                     # label("AxCN2").
P( i(x, i(-x,y)) )                     # label("AxCN3").
end_of_list.

formulas(sos).
-P(i(x,y)) | -P(x) | P(y)          # label("InfConDet").
end_of_list.

formulas(goals).
P ( i(y, i(x,y)) )                          # label("CN18").
P ( i(i(x, i(y,z)), i(y, i(x,z))) )         # label("CN21").
P ( i(i(x, i(y,z)), i(i(x,y), i(x,z))) )    # label("CN35").
P ( i((-(-x)), x) )                         # label("CN39").
P ( i(x, -(-x)) )                           # label("CN40").
P ( i(i(x,y), i(-y, -x)) )                  # label("CN46").
end_of_list.
```

Figure 1. The *prover9* script used to show that CN implies BG. The implementation of condensed detachment is the formula in the "sos" list; substitution is derived from *prover9*'s hyperresolution rule (introduced in the "set" command at the top of the script). CN34, a theorem of [1], has been added to the axioms of [1] to facilitate the derivation. Details of *prover9*'s syntax and semantics can be found in [2].

3.0 Results

Figure 2 shows that CN implies BG.

```
============================== PROOF =================================

% Proof 1 at 0.03 (+ 0.05) seconds: "CN39".

4 P(i(--x,x)) # label("CN39") # label(non_clause) # label(goal).  [goal].
7 P(i(i(x,i(y,i(z,u))),i(i(z,x),i(y,i(z,u))))) # label("CN34").  [assumption].
8 P(i(i(x,y),i(i(y,z),i(x,z)))) # label("AxCN1").  [assumption].
9 P(i(i(-x,x),x)) # label("AxCN2").  [assumption].
10 P(i(x,i(-x,y))) # label("AxCN3").  [assumption].
11 -P(i(x,y)) | -P(x) | P(y) # label("InfConDet").  [assumption].
15 -P(i(--c9,c9)) # label("CN39") # answer("CN39").  [deny(4)].
22 P(i(i(i(-x,y),z),i(x,z))).  [hyper(11,a,8,a,b,10,a)].
26 P(i(i(x,y),i(-y,i(x,z)))).  [hyper(11,a,7,a,b,10,a)].
40 P(i(x,x)).  [hyper(11,a,22,a,b,9,a)].
101 P(i(-x,i(x,y))).  [hyper(11,a,26,a,b,40,a)].
117 P(i(i(i(x,y),z),i(-x,z))).  [hyper(11,a,8,a,b,101,a)].
```

```
343 P(i(--x,x)).  [hyper(11,a,117,a,b,9,a)].
344 $F # answer("CN39").  [resolve(343,a,15,a)].

============================== end of proof ==========================


============================== PROOF =================================

% Proof 2 at 0.72 (+ 0.08) seconds: "CN40".

5 P(i(x,--x)) # label("CN40") # label(non_clause) # label(goal).  [goal].
7 P(i(i(x,i(y,i(z,u))),i(i(z,x),i(y,i(z,u))))) # label("CN34").  [assumption].
8 P(i(i(x,y),i(i(y,z),i(x,z)))) # label("AxCN1").  [assumption].
9 P(i(i(-x,x),x)) # label("AxCN2").  [assumption].
10 P(i(x,i(-x,y))) # label("AxCN3").  [assumption].
11 -P(i(x,y)) | -P(x) | P(y) # label("InfConDet").  [assumption].
16 -P(i(c10,--c10)) # label("CN40") # answer("CN40").  [deny(5)].
22 P(i(i(i(-x,y),z),i(x,z))).  [hyper(11,a,8,a,b,10,a)].
23 P(i(i(x,y),i(i(-x,x),y))).  [hyper(11,a,8,a,b,9,a)].
24 P(i(i(i(i(x,y),i(z,y)),u),i(i(z,x),u))).  [hyper(11,a,8,a,b,8,a)].
26 P(i(i(x,y),i(-y,i(x,z)))).  [hyper(11,a,7,a,b,10,a)].
40 P(i(x,x)).  [hyper(11,a,22,a,b,9,a)].
43 P(i(-i(x,x),y)).  [hyper(11,a,10,a,b,40,a)].
46 P(i(i(x,y),i(-i(z,z),y))).  [hyper(11,a,8,a,b,43,a)].
101 P(i(-x,i(x,y))).  [hyper(11,a,26,a,b,40,a)].
117 P(i(i(i(x,y),z),i(-x,z))).  [hyper(11,a,8,a,b,101,a)].
235 P(i(x,i(-i(y,y),z))).  [hyper(11,a,22,a,b,46,a)].
242 P(i(i(i(-i(x,x),y),z),i(u,z))).  [hyper(11,a,8,a,b,235,a)].
343 P(i(--x,x)).  [hyper(11,a,117,a,b,9,a)].
351 P(i(i(x,y),i(--x,y))).  [hyper(11,a,8,a,b,343,a)].
408 P(i(x,i(---x,y))).  [hyper(11,a,22,a,b,351,a)].
3645 P(i(x,i(y,y))).  [hyper(11,a,242,a,b,9,a)].
3664 P(i(i(i(x,x),y),i(z,y))).  [hyper(11,a,8,a,b,3645,a)].
3722 P(i(i(x,y),i(z,i(x,y)))).  [hyper(11,a,24,a,b,3664,a)].
3740 P(i(x,i(i(-y,y),y))).  [hyper(11,a,3664,a,b,23,a)].
3884 P(i(i(x,i(x,y)),i(z,i(x,y)))).  [hyper(11,a,7,a,b,3722,a)].
5586 P(i(x,i(i(y,i(y,z)),i(y,z)))).  [hyper(11,a,3884,a,b,3884,a)].
6337 P(i(i(x,i(x,y)),i(x,y))).  [hyper(11,a,5586,a,b,5586,a)].
6385 P(i(i(i(x,y),x),i(i(x,y),y))).  [hyper(11,a,24,a,b,6337,a)].
6754 P(i(i(i(i(-x,x),x),y),y)).  [hyper(11,a,6385,a,b,3740,a)].
6798 P(i(i(x,i(-y,y)),i(x,y))).  [hyper(11,a,24,a,b,6754,a)].
6946 P(i(x,--x)).  [hyper(11,a,6798,a,b,408,a)].
6947 $F # answer("CN40").  [resolve(6946,a,16,a)].

============================== end of proof ==========================

============================== PROOF =================================

% Proof 3 at 2.34 (+ 0.08) seconds: "CN18".

1 P(i(y,i(x,y))) # label("CN18") # label(non_clause) # label(goal).  [goal].
7 P(i(i(x,i(y,i(z,u))),i(i(z,x),i(y,i(z,u))))) # label("CN34").  [assumption].
8 P(i(i(x,y),i(i(y,z),i(x,z)))) # label("AxCN1").  [assumption].
9 P(i(i(-x,x),x)) # label("AxCN2").  [assumption].
10 P(i(x,i(-x,y))) # label("AxCN3").  [assumption].
11 -P(i(x,y)) | -P(x) | P(y) # label("InfConDet").  [assumption].
12 -P(i(c1,i(c2,c1))) # label("CN18") # answer("CN18").  [deny(1)].
22 P(i(i(i(-x,y),z),i(x,z))).  [hyper(11,a,8,a,b,10,a)].
23 P(i(i(x,y),i(i(-x,x),y))).  [hyper(11,a,8,a,b,9,a)].
24 P(i(i(i(i(x,y),i(z,y)),u),i(i(z,x),u))).  [hyper(11,a,8,a,b,8,a)].
26 P(i(i(x,y),i(-y,i(x,z)))).  [hyper(11,a,7,a,b,10,a)].
40 P(i(x,x)).  [hyper(11,a,22,a,b,9,a)].
43 P(i(-i(x,x),y)).  [hyper(11,a,10,a,b,40,a)].
46 P(i(i(x,y),i(-i(z,z),y))).  [hyper(11,a,8,a,b,43,a)].
96 P(i(i(i(-x,i(y,z)),u),i(i(y,x),u))).  [hyper(11,a,8,a,b,26,a)].
101 P(i(-x,i(x,y))).  [hyper(11,a,26,a,b,40,a)].
117 P(i(i(i(x,y),z),i(-x,z))).  [hyper(11,a,8,a,b,101,a)].
146 P(i(i(x,i(y,z)),i(i(u,y),i(x,i(u,z))))).  [hyper(11,a,24,a,b,24,a)].
235 P(i(x,i(-i(y,y),z))).  [hyper(11,a,22,a,b,46,a)].
242 P(i(i(i(-i(x,x),y),z),i(u,z))).  [hyper(11,a,8,a,b,235,a)].
343 P(i(--x,x)).  [hyper(11,a,117,a,b,9,a)].
```

```
351 P(i(i(x,y),i(--x,y))).  [hyper(11,a,8,a,b,343,a)].
408 P(i(x,i(---x,y))).  [hyper(11,a,22,a,b,351,a)].
3645 P(i(x,i(y,y))).  [hyper(11,a,242,a,b,9,a)].
3664 P(i(i(i(x,x),y),i(z,y))).  [hyper(11,a,8,a,b,3645,a)].
3722 P(i(i(x,y),i(z,i(x,y)))).  [hyper(11,a,24,a,b,3664,a)].
3740 P(i(x,i(i(-y,y),y))).  [hyper(11,a,3664,a,b,23,a)].
3884 P(i(i(x,i(x,y)),i(z,i(x,y)))).  [hyper(11,a,7,a,b,3722,a)].
5586 P(i(x,i(i(y,i(y,z)),i(y,z)))).  [hyper(11,a,3884,a,b,3884,a)].
6337 P(i(i(x,i(x,y)),i(x,y))).  [hyper(11,a,5586,a,b,5586,a)].
6385 P(i(i(i(x,y),x),i(i(x,y),y))).  [hyper(11,a,24,a,b,6337,a)].
6754 P(i(i(i(i(-x,x),x),y),y)).  [hyper(11,a,6385,a,b,3740,a)].
6798 P(i(i(x,i(-y,y)),i(x,y))).  [hyper(11,a,24,a,b,6754,a)].
6946 P(i(x,--x)).  [hyper(11,a,6798,a,b,408,a)].
6962 P(i(x,i(y,--y))).  [hyper(11,a,3722,a,b,6946,a)].
7348 P(i(i(-x,y),i(-y,x))).  [hyper(11,a,96,a,b,6798,a)].
8124 P(i(i(i(x,--x),y),y)).  [hyper(11,a,6385,a,b,6962,a)].
9635 P(i(x,i(-y,x))).  [hyper(11,a,22,a,b,7348,a)].
9708 P(i(i(x,-y),i(z,i(x,z)))).  [hyper(11,a,146,a,b,9635,a)].
18827 P(i(x,i(y,x))).  [hyper(11,a,8124,a,b,9708,a)].
18828 $F # answer("CN18").  [resolve(18827,a,12,a)].

============================== end of proof ==========================

============================== PROOF =================================

% Proof 4 at 2.48 (+ 0.11) seconds: "CN21".

2 P(i(i(x,i(y,z)),i(y,i(x,z)))) # label("CN21") # label(non_clause) # label(goal).
[goal].
7 P(i(i(x,i(y,i(z,u))),i(i(z,x),i(y,i(z,u))))) # label("CN34").  [assumption].
8 P(i(i(x,y),i(i(y,z),i(x,z)))) # label("AxCN1").  [assumption].
9 P(i(i(-x,x),x)) # label("AxCN2").  [assumption].
10 P(i(x,i(-x,y))) # label("AxCN3").  [assumption].
11 -P(i(x,y)) | -P(x) | P(y) # label("InfConDet").  [assumption].
13 -P(i(i(c3,i(c4,c5)),i(c4,i(c3,c5)))) # label("CN21") # answer("CN21").  [deny(2)].
22 P(i(i(i(-x,y),z),i(x,z))).  [hyper(11,a,8,a,b,10,a)].
23 P(i(i(x,y),i(i(-x,x),y))).  [hyper(11,a,8,a,b,9,a)].
24 P(i(i(i(i(x,y),i(z,y)),u),i(i(z,x),u))).  [hyper(11,a,8,a,b,8,a)].
26 P(i(i(x,y),i(-y,i(x,z)))).  [hyper(11,a,7,a,b,10,a)].
40 P(i(x,x)).  [hyper(11,a,22,a,b,9,a)].
43 P(i(-i(x,x),y)).  [hyper(11,a,10,a,b,40,a)].
46 P(i(i(x,y),i(-i(z,z),y))).  [hyper(11,a,8,a,b,43,a)].
96 P(i(i(i(-x,i(y,z)),u),i(i(y,x),u))).  [hyper(11,a,8,a,b,26,a)].
101 P(i(-x,i(x,y))).  [hyper(11,a,26,a,b,40,a)].
117 P(i(i(i(x,y),z),i(-x,z))).  [hyper(11,a,8,a,b,101,a)].
146 P(i(i(x,i(y,z)),i(i(u,y),i(x,i(u,z))))).  [hyper(11,a,24,a,b,24,a)].
235 P(i(x,i(-i(y,y),z))).  [hyper(11,a,22,a,b,46,a)].
242 P(i(i(i(-i(x,x),y),z),i(u,z))).  [hyper(11,a,8,a,b,235,a)].
343 P(i(--x,x)).  [hyper(11,a,117,a,b,9,a)].
351 P(i(i(x,y),i(--x,y))).  [hyper(11,a,8,a,b,343,a)].
408 P(i(x,i(---x,y))).  [hyper(11,a,22,a,b,351,a)].
3645 P(i(x,i(y,y))).  [hyper(11,a,242,a,b,9,a)].
3664 P(i(i(i(x,x),y),i(z,y))).  [hyper(11,a,8,a,b,3645,a)].
3722 P(i(i(x,y),i(z,i(x,y)))).  [hyper(11,a,24,a,b,3664,a)].
3740 P(i(x,i(i(-y,y),y))).  [hyper(11,a,3664,a,b,23,a)].
3884 P(i(i(x,i(x,y)),i(z,i(x,y)))).  [hyper(11,a,7,a,b,3722,a)].
5586 P(i(x,i(i(y,i(y,z)),i(y,z)))).  [hyper(11,a,3884,a,b,3884,a)].
6337 P(i(i(x,i(x,y)),i(x,y))).  [hyper(11,a,5586,a,b,5586,a)].
6385 P(i(i(i(x,y),x),i(i(x,y),y))).  [hyper(11,a,24,a,b,6337,a)].
6754 P(i(i(i(i(-x,x),x),y),y)).  [hyper(11,a,6385,a,b,3740,a)].
6798 P(i(i(x,i(-y,y)),i(x,y))).  [hyper(11,a,24,a,b,6754,a)].
6913 P(i(i(-x,y),i(i(y,x),x))).  [hyper(11,a,24,a,b,6798,a)].
6946 P(i(x,--x)).  [hyper(11,a,6798,a,b,408,a)].
6962 P(i(x,i(y,--y))).  [hyper(11,a,3722,a,b,6946,a)].
7348 P(i(i(-x,y),i(-y,x))).  [hyper(11,a,96,a,b,6798,a)].
8124 P(i(i(i(x,--x),y),y)).  [hyper(11,a,6385,a,b,6962,a)].
9635 P(i(x,i(-y,x))).  [hyper(11,a,22,a,b,7348,a)].
9708 P(i(i(x,-y),i(z,i(x,z)))).  [hyper(11,a,146,a,b,9635,a)].
18827 P(i(x,i(y,x))).  [hyper(11,a,8124,a,b,9708,a)].
18908 P(i(i(i(x,y),z),i(y,z))).  [hyper(11,a,8,a,b,18827,a)].
20451 P(i(x,i(i(x,y),y))).  [hyper(11,a,18908,a,b,6913,a)].
```

```
20918 P(i(i(x,i(y,z)),i(y,i(x,z)))).  [hyper(11,a,146,a,b,20451,a)].
20919 $F # answer("CN21").  [resolve(20918,a,13,a)].

============================== end of proof ==========================

============================== PROOF =================================

% Proof 5 at 4.84 (+ 0.20) seconds: "CN46".

6 P(i(i(x,y),i(-y,-x))) # label("CN46") # label(non_clause) # label(goal).  [goal].
7 P(i(i(x,i(y,i(z,u))),i(i(z,x),i(y,i(z,u))))) # label("CN34").  [assumption].
8 P(i(i(x,y),i(i(y,z),i(x,z)))) # label("AxCN1").  [assumption].
9 P(i(i(-x,x),x)) # label("AxCN2").  [assumption].
10 P(i(x,i(-x,y))) # label("AxCN3").  [assumption].
11 -P(i(x,y)) | -P(x) | P(y) # label("InfConDet").  [assumption].
17 -P(i(i(c11,c12),i(-c12,-c11))) # label("CN46") # answer("CN46").  [deny(6)].
22 P(i(i(i(-x,y),z),i(x,z))).  [hyper(11,a,8,a,b,10,a)].
23 P(i(i(x,y),i(i(-x,x),y))).  [hyper(11,a,8,a,b,9,a)].
24 P(i(i(i(i(x,y),i(z,y)),u),i(i(z,x),u))).  [hyper(11,a,8,a,b,8,a)].
26 P(i(i(x,y),i(-y,i(x,z)))).  [hyper(11,a,7,a,b,10,a)].
40 P(i(x,x)).  [hyper(11,a,22,a,b,9,a)].
43 P(i(-i(x,x),y)).  [hyper(11,a,10,a,b,40,a)].
46 P(i(i(x,y),i(-i(z,z),y))).  [hyper(11,a,8,a,b,43,a)].
69 P(i(i(i(i(-x,x),y),z),i(i(x,y),z))).  [hyper(11,a,8,a,b,23,a)].
96 P(i(i(i(-x,i(y,z)),u),i(i(y,x),u))).  [hyper(11,a,8,a,b,26,a)].
101 P(i(-x,i(x,y))).  [hyper(11,a,26,a,b,40,a)].
117 P(i(i(i(x,y),z),i(-x,z))).  [hyper(11,a,8,a,b,101,a)].
235 P(i(x,i(-i(y,y),z))).  [hyper(11,a,22,a,b,46,a)].
242 P(i(i(i(-i(x,x),y),z),i(u,z))).  [hyper(11,a,8,a,b,235,a)].
335 P(i(i(i(-x,y),z),i(i(i(x,u),y),z))).  [hyper(11,a,8,a,b,117,a)].
3645 P(i(x,i(y,y))).  [hyper(11,a,242,a,b,9,a)].
3664 P(i(i(i(x,x),y),i(z,y))).  [hyper(11,a,8,a,b,3645,a)].
3722 P(i(i(x,y),i(z,i(x,y)))).  [hyper(11,a,24,a,b,3664,a)].
3740 P(i(x,i(i(-y,y),y))).  [hyper(11,a,3664,a,b,23,a)].
3884 P(i(i(x,i(x,y)),i(z,i(x,y)))).  [hyper(11,a,7,a,b,3722,a)].
5586 P(i(x,i(i(y,i(y,z)),i(y,z)))).  [hyper(11,a,3884,a,b,3884,a)].
6337 P(i(i(x,i(x,y)),i(x,y))).  [hyper(11,a,5586,a,b,5586,a)].
6385 P(i(i(i(x,y),x),i(i(x,y),y))).  [hyper(11,a,24,a,b,6337,a)].
6754 P(i(i(i(i(-x,x),x),y),y)).  [hyper(11,a,6385,a,b,3740,a)].
6798 P(i(i(x,i(-y,y)),i(x,y))).  [hyper(11,a,24,a,b,6754,a)].
7348 P(i(i(-x,y),i(-y,x))).  [hyper(11,a,96,a,b,6798,a)].
34837 P(i(i(i(x,y),z),i(-z,x))).  [hyper(11,a,335,a,b,7348,a)].
34950 P(i(i(x,y),i(-y,-x))).  [hyper(11,a,69,a,b,34837,a)].
34951 $F # answer("CN46").  [resolve(34950,a,17,a)].

============================== end of proof ==========================

============================== PROOF =================================

% Proof 6 at 11.31 (+ 0.95) seconds: "CN35".

3 P(i(i(x,i(y,z)),i(i(x,y),i(x,z)))) # label("CN35") # label(non_clause) # label(goal).
[goal].
7 P(i(i(x,i(y,i(z,u))),i(i(z,x),i(y,i(z,u))))) # label("CN34").  [assumption].
8 P(i(i(x,y),i(i(y,z),i(x,z)))) # label("AxCN1").  [assumption].
9 P(i(i(-x,x),x)) # label("AxCN2").  [assumption].
10 P(i(x,i(-x,y))) # label("AxCN3").  [assumption].
11 -P(i(x,y)) | -P(x) | P(y) # label("InfConDet").  [assumption].
14 -P(i(i(c6,i(c7,c8)),i(i(c6,c7),i(c6,c8)))) # label("CN35") # answer("CN35").
[deny(3)].
22 P(i(i(i(-x,y),z),i(x,z))).  [hyper(11,a,8,a,b,10,a)].
23 P(i(i(x,y),i(i(-x,x),y))).  [hyper(11,a,8,a,b,9,a)].
24 P(i(i(i(i(x,y),i(z,y)),u),i(i(z,x),u))).  [hyper(11,a,8,a,b,8,a)].
26 P(i(i(x,y),i(-y,i(x,z)))).  [hyper(11,a,7,a,b,10,a)].
40 P(i(x,x)).  [hyper(11,a,22,a,b,9,a)].
43 P(i(-i(x,x),y)).  [hyper(11,a,10,a,b,40,a)].
46 P(i(i(x,y),i(-i(z,z),y))).  [hyper(11,a,8,a,b,43,a)].
96 P(i(i(i(-x,i(y,z)),u),i(i(y,x),u))).  [hyper(11,a,8,a,b,26,a)].
101 P(i(-x,i(x,y))).  [hyper(11,a,26,a,b,40,a)].
117 P(i(i(i(x,y),z),i(-x,z))).  [hyper(11,a,8,a,b,101,a)].
146 P(i(i(x,i(y,z)),i(i(u,y),i(x,i(u,z))))).  [hyper(11,a,24,a,b,24,a)].
```

```
235 P(i(x,i(-i(y,y),z))).  [hyper(11,a,22,a,b,46,a)].
242 P(i(i(i(-i(x,x),y),z),i(u,z))).  [hyper(11,a,8,a,b,235,a)].
343 P(i(--x,x)).  [hyper(11,a,117,a,b,9,a)].
351 P(i(i(x,y),i(--x,y))).  [hyper(11,a,8,a,b,343,a)].
408 P(i(x,i(---x,y))).  [hyper(11,a,22,a,b,351,a)].
3645 P(i(x,i(y,y))).  [hyper(11,a,242,a,b,9,a)].
3664 P(i(i(i(x,x),y),i(z,y))).  [hyper(11,a,8,a,b,3645,a)].
3722 P(i(i(x,y),i(z,i(x,y)))).  [hyper(11,a,24,a,b,3664,a)].
3740 P(i(x,i(i(-y,y),y))).  [hyper(11,a,3664,a,b,23,a)].
3884 P(i(i(x,i(x,y)),i(z,i(x,y)))).  [hyper(11,a,7,a,b,3722,a)].
5586 P(i(x,i(i(y,i(y,z)),i(y,z)))).  [hyper(11,a,3884,a,b,3884,a)].
6337 P(i(i(x,i(x,y)),i(x,y))).  [hyper(11,a,5586,a,b,5586,a)].
6385 P(i(i(i(x,y),x),i(i(x,y),y))).  [hyper(11,a,24,a,b,6337,a)].
6754 P(i(i(i(i(-x,x),x),y),y)).  [hyper(11,a,6385,a,b,3740,a)].
6798 P(i(i(x,i(-y,y)),i(x,y))).  [hyper(11,a,24,a,b,6754,a)].
6913 P(i(i(-x,y),i(i(y,x),x))).  [hyper(11,a,24,a,b,6798,a)].
6946 P(i(x,--x)).  [hyper(11,a,6798,a,b,408,a)].
6962 P(i(x,i(y,--y))).  [hyper(11,a,3722,a,b,6946,a)].
7348 P(i(i(-x,y),i(-y,x))).  [hyper(11,a,96,a,b,6798,a)].
8124 P(i(i(i(x,--x),y),y)).  [hyper(11,a,6385,a,b,6962,a)].
9635 P(i(x,i(-y,x))).  [hyper(11,a,22,a,b,7348,a)].
9708 P(i(i(x,-y),i(z,i(x,z)))).  [hyper(11,a,146,a,b,9635,a)].
18827 P(i(x,i(y,x))).  [hyper(11,a,8124,a,b,9708,a)].
18908 P(i(i(i(i(x,y),z),i(y,z))).  [hyper(11,a,8,a,b,18827,a)].
20451 P(i(x,i(i(x,y),y))).  [hyper(11,a,18908,a,b,6913,a)].
20918 P(i(i(x,i(y,z)),i(y,i(x,z)))).  [hyper(11,a,146,a,b,20451,a)].
45144 P(i(i(x,y),i(i(z,x),i(z,y)))).  [hyper(11,a,20918,a,b,8,a)].
55392 P(i(i(x,i(y,z)),i(i(x,y),i(x,z)))).  [hyper(11,a,7,a,b,45144,a)].
55393 $F # answer("CN35").  [resolve(55392,a,14,a)].

============================== end of proof ==========================
```

Figure 2. Summary of a *prover9* ([2]) proof showing that CN ([1]) implies BG ([10]).

The total time to complete the proofs shown in Figure 2 was ~23 seconds on the platform described in Section 2.0.

4.0 Conclusions and discussion

Section 3 demonstrates that CN implies BG. A companion paper ([16]) proves BG implies CN. The proof in Figure 2 appears to be novel.

There are some interesting relationships among various subproofs of the proofs of CN18, CN21, CN35, CN40, and CN46. In particular, the proofs of CN18, CN21, CN35, and CN40 share a subproof consisting of Lines 22-6946; this shared subproof can thus be regarded as a common set of lemmas for CN18, CN21, CN35, and CN40. Lines 7-6946 of the proof of CN40 (all but two lines of that proof) are a subproof of the proof of CN18 and thus may be regarded as a set of lemmas for the proof of CN18.

5.0 References

[1] Łukasiewicz J. *Elements of Mathematical Logic*. Second Edition (1958). Trans. by Wojtasiewicz O. Pergamon Press. 1963.

[2] McCune WW. *prover9 and mace4.* http://www.cs.unm.edu/~mccune/prover9/. 2009.

[3] Aristotle. *Prior Analytics*. Trans. by A. J. Jenkinson. In Aristotle. *The Basic Works*

of Aristotle. Ed. by R. McKeon. Random House. 1941. pp. 62-107.

[4] Aristotle. *Posterior Analytics.* Trans. by G. R. G. Mure. In Aristotle. *The Basic Works of Aristotle.* Ed. by R. McKeon. Random House. 1941. pp. 108-186.

[5] Tarski A. *Introduction to Logic*. Trans. by O. Helmer. Dover. 1941.

[6] Hempel C. Studies in the logic of explanation. In Hempel C. *Aspects of Scientific Explanation and Other Essays in the Philosophy of Science.* Free Press. 1965. pp. 245-290.

[7] Quine WVO. *Philosophy of Logic*. Second Edition. Harvard. 1986.

[8] Chang CC and Keisler HJ. *Model Theory*. North-Holland. 1990.

[9] Russell B and Whitehead AN. *Principia Mathematica*. Volume I (1910). Merchant Books. 2009.

[10] Frege G. Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens. Halle. 1879. Translated in van Heijenoort J. Begriffsschrift, a formula language, modeled upon that of arithmetic, for pure thought. *From Frege to Gödel: A Source Book in Mathematical Logic, 1879-1931*. Harvard. 1967. pp. 3-82.

[11] Horn A. On sentences which are true of direct unions of algebras. *Journal of Symbolic Logic* 16 (1951), 14–21.

[12] Church A. *Introduction to Mathematical Logic.* Volume I. Princeton. 1956.

[13] Birkhoff G and von Neumann J. The logic of quantum mechanics. *Annals of Mathematics* 37 (1936), 823-843.

[14] Kant I. *Kant's Introduction to Logic* (1800). Trans. by Abbott TK. Greenwood Press. 1963.

[15] Cohen MR and Nagel E. *An Introduction to Logic and Scientific Method.* Harcourt, Brace, and Company. 1934.

[16] Horner JK. An automated derivation of Łukasiewicz's sentential calculus from Frege's. *Proceedings of the 2011 International Conference on Artificial Intelligence.* CSREA Press. Forthcoming.

Honey Bee Mating Optimization Based LFC Design in a Deregulated Power System

H. Shayeghi
Technical Eng. Department
University of Mohaghegh Ardabili
Ardabil, Iran

H. A. Shayanfar
Electrical Eng. Department, Islamic
Azad University, South Tehran Branch,
Tehran, Iran

A. Ghasemi
Technical Eng. Department
University of Mohaghegh Ardabili
Ardabil, Iran

hshayeghi@gmail.com, hashayanfar@yahoo.com, ghasemi.agm@gmail.com

Abstract- *In this paper, optimal tuning of the PID controller for solution of the Load frequency Control (LFC) problem in the restructured power system is proposed using Honey Bee Mating Optimization (HBMO) technique. The problem of robustly tuning of PID based LFC design is formulated as an optimization problem according to the time domain-based objective function which is solved by the HBMO technique that has a strong ability to find the most optimistic results. To ensure performance and robustness of the proposed control strategy to stabilize frequency oscillations, the design process takes a wide range of operating conditions and system nonlinearities into account. To demonstrate the effectiveness of the proposed method a two-area restructured power system is considered as a test system under different operating conditions. The simulation results are shown to maintain robust performance in comparison with the particle swarm optimization and genetic algorithm based tuned PID controllers through some performance indices. Results evaluation show that the proposed control strategy achieves good robust performance for wide range of system parameters and load changes in the presence of system nonlinearities and is superior to the other controllers.*

Keywords: Load Frequency Control, HBMO, Restructured Power System, PID Control.

1. Introduction

Global analysis of the power system markets shows that the frequency control is one of the most profitable ancillary services at these systems. This service is related to the short-term balance of energy and frequency of the power systems. The most common methods used to accomplish frequency control are generator governor response (primary frequency regulation) and Load Frequency Control (LFC). The goal of the LFC is to reestablish primary frequency regulation capacity, return the frequency to its nominal value and minimize unscheduled tie-line power flows between neighboring control areas [1].

An industrial process, such as a power system, contains different kinds of uncertainties due to changes in system parameters and characteristics, loads variation and errors in the modeling. On the other hand, the operating points of a power system may change very much randomly during a daily cycle. Because of this, a fixed controller based on classical theory is certainly not suitable for the LFC problem. Thus, some authors have suggested variable structure [2-3], neural networks methods [4-5] and robust control methodologies [6-8] for the solution of the LFC problem to deal with parameter variations. Although via these methods, the parameter variation is considered to the synthesis; but, models of large scalar power system have several features that preclude direct application of these methodologies. Among these properties, the most prominent are: very large (and unknown) model order, uncertain connection between subsystems, broad parameter variation and elaborate organizational structure.

Despite the potential of the modern control techniques with different structure, Proportional Integral Derivative (PID) type controller is still widely used for the solution of the LFC problem [9-11]. This is because it performs well for a wide class of process. Also, they give robust performance for a wide range of operating conditions and easy to implement. On the other hand, Shayeghi *et. al* [9] have presented a comprehensive analysis of the effects of different PID controller parameters on the overall dynamic performance of the LFC problem. It is shown that the appropriate selection of the PID controller parameters results in a satisfactory performance during system upsets. Thus, the optimal tuning of a PID gains is required to get the desired level of robust performance. Since the optimal setting of PID controller gains is a *multimodal* optimization problem (i.e., there exists more than one local optimum) and more complex due to the nonlinearity, complexity and

* Correspanding Author. E-Mail Address: hashayanfar@yahoo.com (H. A. Shayanfar)

time-variability of the real world power systems operation. Hence, local optimization techniques, which are well elaborated upon, are not suitable for such a problem.

Recently, global optimization techniques like Genetic Algorithms (GA), Particle Swarm Optimization (PSO) and simulated annealing [12-14] have been applied for optimal tuning of the PID based LFC schemes. These evolutionary algorithms are heuristic population-based search procedures that incorporate random variation and selection operators. Although, these methods seem to be good methods for the solution of the PID parameter optimization problem, however, when the system has a highly epistatic objective function (i.e. where parameters being optimized are highly correlated), and the number of parameters to be optimized is large, then they have degraded efficiency to obtain global optimum solution. In order to overcome these drawbacks, a Honey Bee Mating Optimization (HBMO) based PID type controller is proposed for the solution of the LFC problem in this paper. The HBMO algorithm is a typical swarm-based approach to optimization, in which the search algorithm is inspired by the honey-bee mating process [15] and has emerged as a useful tool for the engineering optimization. It simulates the mating process of the queen of the hive. The mating process of the queen begins when the queen flights away from the nest performing the mating flight during which the drones follow the queen and mate with her in the air. The HBMO algorithm is a hybrid evolutionary algorithm which is comprised of GA, SA, local search, and some innovations for its self-adaptation. It has also been found to be robust in solving problems featuring non-linearity, non-differentiability and high dimensionality [16]. Unlike the other heuristic techniques, The HBMO algorithm has a flexible and well-balanced mechanism to enhance the global and local exploration abilities.

In this study, the problem of robust PID based load frequency controller design is formulated as an optimization problem. The controller is automatically tuned with optimization of a time domain based objective function by HBMO such that the relative stability is guaranteed and the time domain specifications concurrently secured. The effectiveness of the proposed controller is demonstrated in comparison with the designed controller using PSO and GA techniques through time domain simulation studies and some performance indices to damp frequency oscillations under different operating conditions and system nonlinearities.

2. Power System Model

The LFC problem has been dealt with extensively for more than four decades. A comprehensive literatures review about the earlier studied in the field of LFC problem has been presented by Shayeghi et. al [1]. Generalized dynamical model for the LFC scheme has been developed in Ref. [17] based on the possible contracts in the deregulated environments. This section gives a brief overview on this generalized model. In the deregulated power system, Generation Companies (GENCOs) may or may not participate in the LFC task and Distribution Companies (DISCOs) have the liberty to contract with any available GENCOs in their own or other areas. The concept of an Augmented Generation Participation Matrix (AGPM) is introduced to express these possible contracts in the generalized model. For example, the AGPM structure for a large scale power system with N control area is given by:

$$AGPM = \begin{bmatrix} AGPM_{11} & \cdots & AGPM_{1N} \\ \vdots & \ddots & \vdots \\ AGPM_{N1} & \cdots & AGPM_{NN} \end{bmatrix} \quad (1)$$

Where,

$$AGPM_{ij} = \begin{bmatrix} gpf_{(s_i+1)(z_j+1)} & \cdots & gpf_{(s_i+1)(z_j+m_j)} \\ \vdots & \ddots & \vdots \\ gpf_{(s_i+n_i)(z_j+1)} & \cdots & gpf_{(s_i+n_i)(z_j+m_j)} \end{bmatrix}$$

For i, j=1,…,N *and*

$$s_i = \sum_{k=1}^{i-1} n_i \,,\; z_j = \sum_{k=1}^{j-1} m_j \,, \quad s_1 = z_1 = 0$$

Where, n_i and m_i are the number of GENCOs and DISCOs in area i and gpf_{ij} refer to '*generation participation factor*' and shows the participation factor GENCO i in total load following requirement of DISCO j based on the possible contract. The sum of all entries in each column of AGPM is unity. To illustrate the effectiveness of the modeling strategy and proposed control design, a two control area power system is considered as a test system. It is assumed that each control area includes two GENCOs and DISCOs. Block diagram of the generalized LFC scheme for test power system is shown in Fig. 1. The power system parameters are given in the Appendix.

The dotted and dashed lines show the demand signals based on the possible contracts between GENCOs and DISCOs which carry information as to which GENCO has to follow a load demanded by which DISCO. These new information signals were absent in the traditional LFC scheme. As there are many GENCOs in each area, ACE signal has to be distributed among them due to their ACE participation factor in the LFC task and $\sum_{j=1}^{n_i} apf_{ji} = 1$. We can write [17]:

$$d_i = \Delta P_{Loc,j} + \Delta P_{di}, \Delta P_{Loc,j} = \sum_{j=1}^{m_i} \Delta P_{Lj-i}, \Delta P_{di} = \sum_{j=1}^{m_i} \Delta P_{ULj-i} \quad (2)$$

$$\eta_i = \sum_{j=1 \& j \neq i}^{N} T_{ij} \Delta f_j \quad (3)$$

$$\zeta_i = \Delta P_{tie,i,sch} = \sum_{k=1 \& k \neq i}^{N} \Delta P_{tie,ik,sch} \quad (4)$$

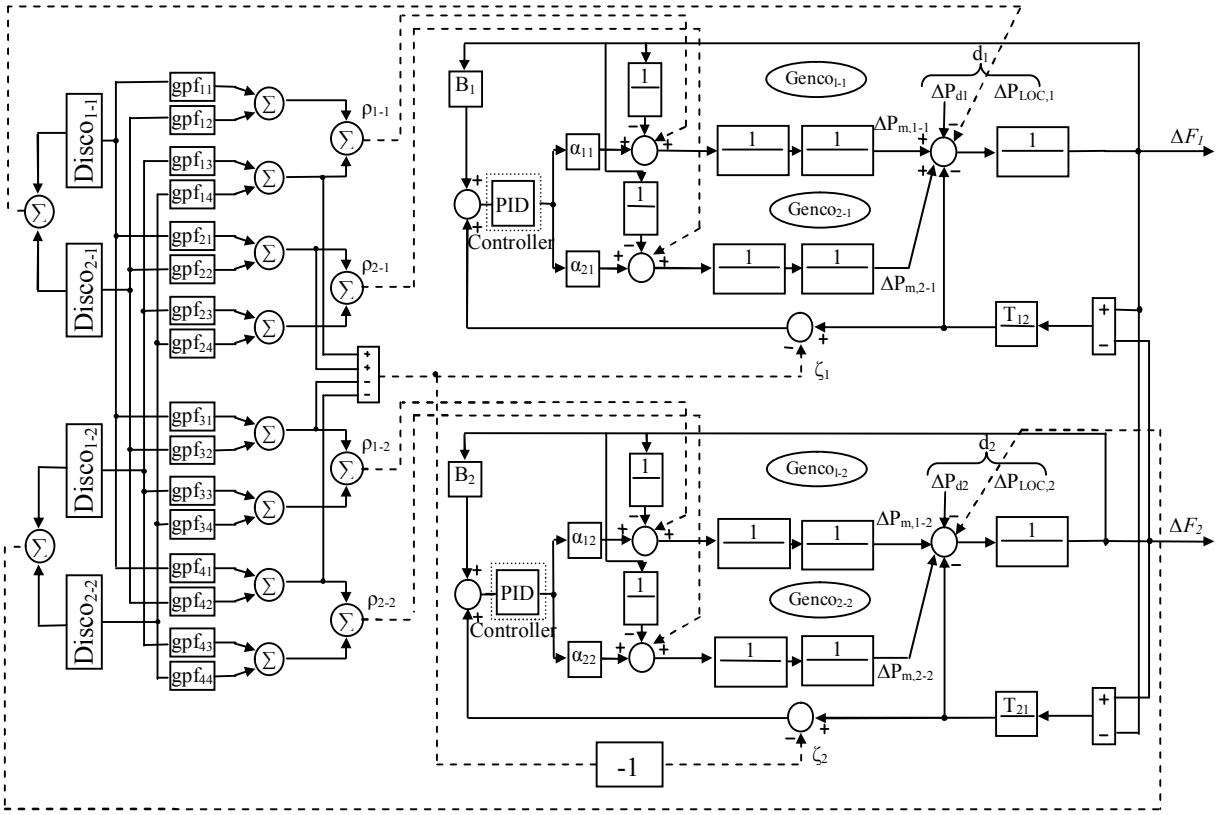


Fig. 1. The generalized LFC scheme in the restructured system

$$\Delta P_{tie,ik,sch} = \sum_{j=1}^{n_i}\sum_{t=1}^{m_k} gpf_{(s_i+j)(z_k+t)}\Delta P_{Lt-k} - \sum_{t=1}^{n_k}\sum_{j=1}^{m_i} gpf_{(s_k+t)(z_i+j)}\Delta P_{Lj-i} \quad (5)$$

$$\rho_i = [\rho_{1i} \quad \cdots \quad \rho_{ki} \quad \cdots \quad \rho_{n_i i}] \quad (6)$$

$$\rho_{ki} = \sum_{j=1}^{N}[\sum_{t=1}^{m_j} gpf_{(s_i+k)(z_j+t)}\Delta P_{Lt-j}]$$

$$\Delta P_{m,k-i} = \rho_{ki} + apf_{ki}\Delta P_{di} \quad (7)$$

Where, $\Delta P_{m,ki}$ is the desired total power generation of a GENCO k in area i and must track the demand of the DISCOs in contract with it in the steady state.

One of the importance constraints in the LFC problem is GRC, i.e. practical limit on the rate of change in the generation power of each generator. The results in [11, 18] indicated that the GRC would influence the dynamic responses of the system significantly and lead to larger overshoot and longer settling time. In order to take effect of the GRC into account, the linear model of turbine $\Delta PV_i/\Delta PT_i$ in Fig. 1 is usually replaced by a nonlinear model of Fig. 2 (with $\pm\delta$ limit). Also, a limiter, bounded by $\pm\delta$ limit was used within the PID controller for governor system to prevent the excessive control action. In this study, δ is considered to be 0.2 [13].

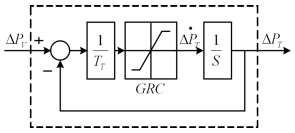


Fig. 2. Nonlinear turbine

3. HBMO

A honey-bee colony typically consists of a single egg laying long-lived queen, several thousand drones (depending on the season), workers and a large family of bees living in one bee-hive [19]. Each bee undertakes sequences of actions which unfold according to genetic, ecological and social condition of the colony.

A mating flight starts with a dance performed by the queen who then starts a mating flight during which the drones follow the queen and mate with her in the air. After the mating process, the drones die. In each mating, sperm reaches the spermatheca and accumulates there to form the genetic pool of the colony. Each time a queen lays fertilized eggs, she randomly retrieves a mixture of the sperm accumulated in the spermatheca to fertilize the egg and this task can only be done by the queen [15-16].

The HBMO algorithm combines different phases of the marriage process of the honey bee. It starts with random generation of a set of initial solutions. Based on their fitness, randomly generated solutions are then ranked. The fittest solution is named queen, whereas the

remaining solutions are categorized as drones (i.e., trial solutions). In order to form the hive and start mating process, the queen, drones and workers (predefined heuristic functions) should be defined. Each queen is characterized with a genotype, speed, energy and a spermatheca with defined capacity. In the next step, drones must be nominated to mate with the queen probabilistically during the mating flight. At the start of the flight, the queen is initialized with some energy content and returns to her nest when the energy is within some threshold of either near zero or when the spermatheca is full. The mating flight may be considered as a set of transitions in a state-space (the environment). An annealing function is used to describe the probability of a Drone (D) that successfully mates with the Queen (Q) as [15]:

$$prob(Q,D) = e^{\frac{-\Delta(f)}{S(t)}} \quad (8)$$

Where, $\Delta(f)$ is the absolute difference of the fitness of D and the fitness of Q and the $S(t)$ is the speed of queen at time t. The fitness of the resulting chromosomes of drone, queen or brood is determined by evaluating the value of the objective function. After each transition in space, the queen's speed and energy decays is given by:

$$S(t+1) = \alpha \times S(t) \quad (9)$$

$$E(t+1) = E(t) - \gamma \quad (10)$$

Where, $\alpha(t)$ is the speed reduction factor and γ is the amount of energy reduction after each transition (α , $\gamma \in [0,1]$).

In order to develop the algorithm, the capability of workers is restrained in brood care and thus, each worker may be regarded as a heuristic that acts, to improve and/or take care of a set of broods. The rate of improvement in the brood's genotype, defines the heuristic fitness value. The fitness of the resulting genotype is determined by evaluating the value of the objective function of the brood genotype and/or its normalized value. It is important to note that a brood has only one genotype [20].

In general, the whole process of HBMO algorithm as can be summarized at the five main steps as follows:

i) Generate the initial drone sets and queen: The algorithm starts with the mating flight, where a queen (best solution) selects drones probabilistically to form the spermatheca (list of drones). A drone then selected from the list randomly for the creation of broods.

ii) Flight matting: This steps do the flight matting of queen Q. The best drone D_k with the largest *prob(Q, D)* among the drone set D is selected the object of matting for the queen Q. After the flight matting the queen's speed and energy decay is reduced by Eq. (9). The flight matting is continues until the speed $S(t)$ is less than a threshold d or the number of sperms of the queen's spermatheca is less than the one threshold.

iii) Breeding process: In this step, a population of broods is generated based on matting between the queen and the drones stored in the queen's spermatheca. The breeding process can transfer the genes of drones and the queen to the j-th individual based on the Eq. (11).

$$child = parent1 + \beta(parent2 - parent1) \quad (11)$$

Where β is the decreasing factor ($\beta \in [0,1]$).

iv) Adaptation of worker's fitness: The population of broods is improved by applying the mutation operators as follows:

$$Brood_i^k = Brood_i^k \pm (\delta + \varepsilon) Brood_i^k \quad (12)$$
$$\delta \in [0,1], 0 < \varepsilon < 1$$

The δ is randomly generated and ε is predefined. The best brood ($brood_{best}$) with maximum objective function value is selected as the candidate queen. If the objective function of $brood_{best}$ is superior to the queen, the queen replace with $brood_{best}$.

v) Check the termination criteria: If the termination criteria satisfied finish the algorithm, else generate new drones set and go to step 2.

The algorithm continues with three user-defined parameters and one predefined parameter. The predefined parameter is the number of Workers (W), representing the number of heuristics encoded in the program [15, 19]. The user-defined parameters are number of queens, the queen's spermatheca size representing the maximum number of mating per queen in a single mating flight and the number of broods that will be born by all queens. The speed of each queen at the start of each mating flight initialized at randomly. Since this algorithm is the combination of simulated annealing, genetic operator and swarm intelligence, it is very interesting optimization algorithm that used in optimization problems of reservoir operation.

4. HBMO Based LFC Design

Nowadays, despite the significant developments of recent years in control theory and technology PID controllers are used in almost all sectors of industry and science such as power systems [10, 13]. This is because it performs well for a wide class of process. Also, they give robust performance for a wide range of operating conditions. Furthermore, they are easy to implement using analogue or digital hardware and familiar to engineers. In this study, PID controller is used for the solution of LFC problem. It should be noted that the transient performance of the power system with respect to the control of the frequency and tie-line power flows obviously depends on the optimal tuning of the PID controller's parameters. It is well know that the conventional methods to tune PID gains not able to locate or identify the global optimum for achieving the

desired level of system robust performance due to the complexity and multi-variable conditions of the power systems and also, they may be tedious and time consuming. In order to overcome these drawbacks and provide optimal control performance, HBMO algorithm is proposed to off-line optimal tune of the PID gains under different operating conditions. Figure 4 shows the block diagram of HBMO based tuned PID controller to solve the LFC problem for each control area (Fig. 1).

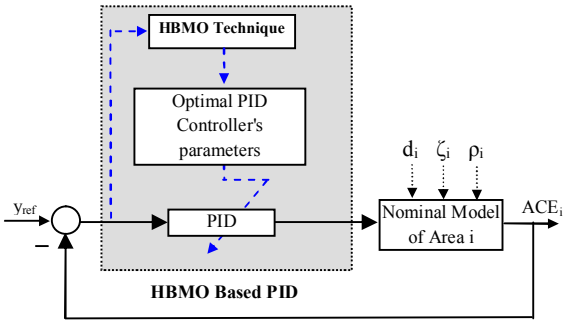


Fig. 4. The proposed HBMO based PID controller structure.

By taking ACE_i as the system output, the control vector for the PID controller in each control area is given by:

$$u_i = K_{Pi} ACE_i + K_{Ii} \int ACE_i dt + K_{Di} \dot{ACE}_i \quad (13)$$

The gains K_{Pi}, K_{Ii} and K_{Di} are tuned using HBMO technique and then, the PID controller generates the control signal that applies to the governor set point in each area. In this study, the ABC module works offline.

Simulation results and eigenvalue analysis show that the open loop system performance is affected more significantly by changing in the K_{pi}, T_{pi}, B_i and T_{ij} than changes of other parameters [18]. Thus, to illustrate the capability of the proposed strategy, in the view point of uncertainty our focus will be concentrated on variation of these parameters. It should be noted that the choice of the properly objective function is very important in synthesis procedure for achieving the desired level of system robust performance. Because, different objective functions promote different HBMO behaviors, which generate fitness value providing a performance measure of the problem considered. For our optimization problem, an Integral of Time multiplied Absolute value of the Error (ITAE) is taken as the objective function. The objective function is defined as follows:

$$J = Max\left\{ITAE^{p=-\%30}, ITAE^{p=-\%20}, ..., ITAE^{p=+\%30}\right\} \quad (14)$$

$$ITAE^{p} = \int_0^{t=tsim} t\,|ACE_i|dt \quad (15)$$

Where, t_{sim} is the time range of simulation; N is the number of area control in power systems and p is percent value of the uncertain plant parameters from nominal values for which the optimization is carried out. For the objective function calculation, the time-domain simulation of the power system model is carried out for the simulation period. It is aimed to minimize this objective function in order to improve the system response in terms of the settling time and overshoots. The design problem can be formulated as the following constrained optimization problem, where the constraints are the PID controller parameter bounds.

Minimize J Subject to

$$K_{Pi}^{\min} \le K_{Pi} \le K_{Pi}^{\max} \quad (16)$$
$$K_{Ii}^{\min} \le K_{Ii} \le K_{Ii}^{\max}$$
$$K_{Di}^{\min} \le K_{Di} \le K_{Di}^{\max}$$

Typical ranges of the optimized parameters are [0.01-20]. To improve the overall system dynamical performance in a robust way and optimization synthesis, this paper employs HBMO technique to solve the above optimization problem and search for the optimal or near optimal set of PID controller parameters (K_{Pi}, K_{Ii} and K_{Di} for i=1, 2 ,..., N).

The optimization of PID controller parameters is carried out by evaluating the objective cost function as given in Eq. (16), which considers a multiple of operating conditions. Consider that all DISCOs contract with available GENCOs for power as per the following AGPM. All GENCOs participate in the LFC task.

$$AGPM_2 = \begin{bmatrix} 0.5 & 0.25 & 0 & 0.3 \\ 0.2 & 0.25 & 0 & 0 \\ 0 & 0.25 & 1 & 0.7 \\ 0.3 & 0.25 & 0 & 0 \end{bmatrix}$$

$$apf_{11} = 0.75, apf_{21} = 0.25, apf_{12} = apf_{22} = 0.5$$

It is assume that a large step load 0.1 puMW is demanded by each DISCOs in all areas. Moreover, consider that DISCOs of area 1 and 2 demand 0.05 and 0.03 puMW of excess power, respectively, which is reflected as a local load perturbation of the areas. Under this contracted scenario, the operating conditions are obtained with variation system of K_{pi}, T_{pi}, B_i and T_{ij} from -25% to 25% of the nominal values by 5% step (i.e. 11 operating conditions. It should be noted that HBMO algorithm is run several times and then, optimal set of PID controller parameters is selected. The final values of the optimized parameters with objective function, J, using HBMO, PSO [13] and GA techniques are given in Table 1.

Table 1. Optimized parameters of PID controller

Method	PID Gains					
	Area 1			Area 2		
	KP	KI	KD	KP	KI	KD
HBMO	2.62	3.95	1.32	1.83	2.09	2.63
PSO	0.88	2.05	0.28	0.01	0.69	0.16
GA	1.18	1.05	0.22	1.08	0.56	1.10

5. Simulation Results

The test system for LFC as shown in Fig. 1 consists of two areas control, and its parameters are given in the Appendix A. The considered system is controlled by using: 1) the PSO based tuned PID controller (PSOPID)

[13]; 2) the GA based tuned PID controller (GAPID); and 3) the HBMO based PID controller designed according to the procedure described in section 4.

A case of combined poolco and bilateral based contracts between DISCOs and available GENCOs is considered based on the AGPM as given in pervious section. It is assumed that a large step load 0.1 puMW is demanded by all DISCOs. Moreover, consider that DISCOs of area 1 and 2 demands 0.05 and 0.03 puMW of excess power, respectively, which is reflected as a local load perturbation of the areas. Based on the given AGPM all GENCOs participate in the LFC task. Power system responses with 25% increase and decrease in uncertain parameters K_{Pi}, T_{Pi}, B_i and T_{ij} are depicted in Fig. 5 and 6, respectively. Using the proposed HBMO based method, the frequency deviation of all areas is quickly driven back to zero and the tie-line power flows properly converges to the specified values in the steady state. i.e.: $\Delta P_{tie,\,12,sch}$=*-0.05 puMW*.

To demonstrate performance robustness of the proposed method, two performance indices: the Integral of the Time multiplied Absolute value of the Error (ITAE) and Figure of Demerit (FD) based on the system performance characteristics are defined as [21]:

$$FD = (OS(dF_1)\times 1000)^2 + (US(dF_1)\times 1000)^2 + (T_s(dF_1))^2 \quad (17)$$

$$ITAE(\Delta f) = 150\int_0^{t_{sim}} t\,dF_1\,dt \quad (18)$$

$$ITAE(\Delta p) = 150\int_0^{t_{sim}} t\,dP_{tie,12}\,dt \quad (19)$$

It is worth mentioning that the lower the value of these indices is, the better the system response in terms of time-domain characteristics. Numerical results of performance robustness for all cases as given in Table 1 for scenario 1 and 2 are listed in Tables 3 and 4, respectively.

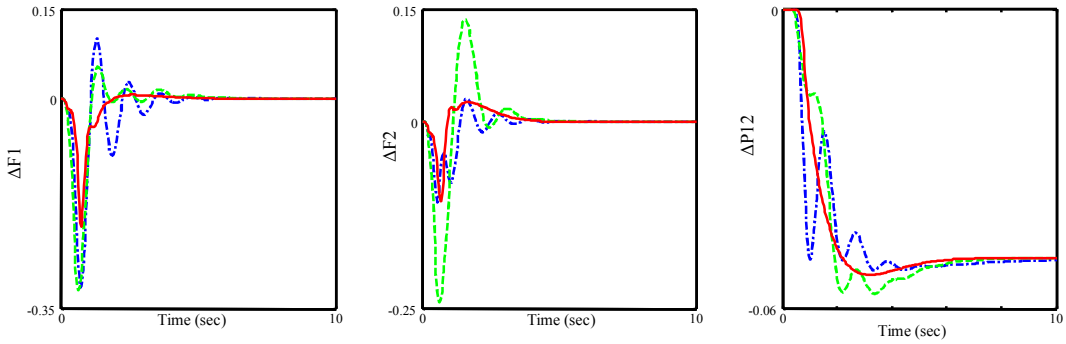


Fig. 5. The performance of the controllers; Solid (HBMO), Dashed (PSO), Dashed-dotted (GA).

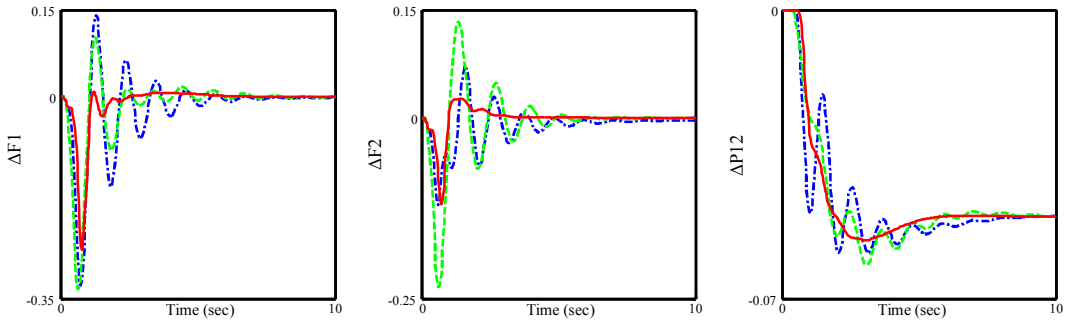


Fig. 6. The performance of the controllers; Solid (HBMO), Dashed (PSO), Dashed-dotted (GA).

Change of Parameters	ITAE(Δ*f*)			ITAE(Δ*P12*)			FD/10^5		
	HBMO	PSO	GA	HBMO	PSO	GA	HBMO	PSO	GA
25%	26.2992	37.0005	60.0152	375.9531	376.8340	369.3946	0.44481	1.0426	1.2353
20%	26.3995	37.8487	60.9840	376.2713	377.0657	369.5786	0.46378	1.0626	1.2464
15%	26.4203	39.3466	63.1623	376.5880	377.2907	369.4395	0.48277	1.0824	1.2590
10%	26.5430	40.7552	65.5567	376.9255	377.4618	369.1992	0.50167	1.1016	1.2682
5%	26.4085	42.1312	63.9013	377.2577	377.5862	370.0499	0.52079	1.1201	1.2781
Nominal	26.7240	43.4446	65.0616	377.3659	377.6599	370.1974	0.54779	1.1385	1.2868
-5%	27.1932	44.7471	66.3566	377.3352	377.6788	370.3431	0.57139	1.1561	1.2937
-10%	25.3113	45.9424	68.3185	377.4995	377.6427	370.4644	0.60114	1.1724	1.2989
-15%	24.9030	48.1279	70.7752	377.4958	377.5621	370.6385	0.63011	1.1876	1.3050
-20%	23.8757	51.8110	74.7198	377.3543	377.4533	370.8449	0.65761	1.2019	1.3082
-25%	24.2113	57.5140	80.3404	377.2707	377.3295	371.1082	0.68522	1.2146	1.3069

It can be seen that the values of these system performance characteristics with the proposed HBMO based tuned controller are much smaller compared to that of PSO and GA based designed PSS. This demonstrates that the overshoot, undershoot, settling time and speed deviations of machine is greatly reduced by applying the proposed HBMO based tuned load frequency controller.

6. Conclusions

In this paper**,** the ABC algorithm has been successfully applied to the robust design of PID controllers for the solution of the LFC problem in the restructured power system. The design problem of the robustly selecting controller parameters is converted into an optimization problem according to time domain-based objective function over a wide range of operating conditions which is solved by the HBMO technique which is a very simple, robust and population based stochastic optimization algorithm. The effectiveness of the proposed strategy was tested on a two-area restructured power system under possible contracts with various load changes in the presence of modeling uncertainties and GRC. The simulation results show that the proposed HBMO based tuned PID controller achieves good robust performance for a wide range of system parameters and is superior to PSO and GA based tuned PID controllers. The system performance characteristics in terms of 'ITAE' and 'FD' indices reveal that the proposed robust PID type tuned controller is a promising control scheme for the solution of the LFC problem.

Appendix: System Data

$$T_{T11} = T_{T21} = T_{T12} = T_{T22} = 0.3\,\text{sec}$$
$$T_{G11} = T_{G21} = T_{G12} = T_{G22} = 0.3\,\text{sec}$$
$$T_{G11} = T_{G21} = T_{G12} = T_{G22} = 0.3\,\text{sec}$$
$$R_{11} = R_{21} = R_{12} = R_{22} = 2.4Hz\ /\ pu$$
$$K_{p1} = 20, K_{p2} = 120; T_{p1} = 20, T_{p2} = 120,$$
$$B_1 = B_2 = 0.425(pu\ /\ Hz), T_{12} = 0.54(pu\ /\ Hz)$$

References

[1] H. Shayeghi, H. A. Shayanfar, A. Jalili, "Load frequency control strategies: A state- of-the-art survey for the researcher", Energy Conversion and Management, Vol. 49, pp.344-353, 2008.

[2] H. Jiang, H. Cai, F. Dorsey and Z. Qu, "Toward a globally robust decentralized control for large-scale power systems", IEEE Trans. on Control System Technology, Vol. 5, No.3, pp. 309-319, 1997.

[3] Z. M. Al-Hamouz, H. N. Al-Duwaish, "A new load frequency variable structure controller using genetic algorithms", Electric Power Systems Research, Vol. 55, No. 1, pp. 1-6, 2000.

[4] H. L. Zeynelgil, A. Demiroren, N. S. Sengor, "The Application of ANN technique to automatic generation control for multi-area power system", Electrical Power and Energy Systems, Vol. 24, pp. 545-354, 2002.

[5] H. Shayeghi, H. A. Shayanfar, "Application of ANN technique based on μ-synthesis to load frequency control of interconnected power system", Electrical Power & Energy Systems, Vol. 28, pp. 503-511, 2006.

[6] H. Shayeghi, "A robust decentralized power system load frequency control", Journal of Electrical Engineering, Vol. 59, No. 6, pp. 281-293, 2008.

[7] M. Karrari, H. Shayeghi, M. Abedi, M. B. Menhaj, "Design of H_∞ controller load frequency control in electrical power systems", Amirkabir Journal of Science & Technology, Vol. 11, No. 41, pp. 79-88, 1999.

[8] H. Bevrani, Y. Mitani, K. Tsuji, "Sequential design of decentralized load frequency controllers using μ synthesis and analysis, *Energy Conversion and Management*, Vol. 45, No. 6, pp. 865-881, 2004.

[9] S. P. Ghoshal, Optimizations of PID gains by particle swarm optimizations in fuzzy based automatic generation control", Electric Power Systems Research, Vol. 70, No. 3, pp. 203-212, 2004.

[10] A. Khodabakhshian, M. Edrisi, "A new robust PID load frequency controller", Control Engineering Practice, Vol. 16, pp. 1069-1080, 2008.

[11] W. Tan, "Tuning of PID load frequency controller for power systems", Energy Conversion and Management, Vol. 50, pp. 1465-1472, 2009.

[12] S. K. Aditya, D. Das, "Design of load frequency controllers using genetic algorithm for two-area interconnected hydro power system", Electrical Power Component Systems, Vol. 31, No. 1, pp. 81-94, 2003.

[13] G. Zarighavar, H. Shayeghi, A. Sakhavati, S. H. Hosseini, "PID type LFC design consideration uncertainties and nonlinearities in power system using PSO", Proc. of the 24th International Power Systems Conference, pp. 1-6, Tehran, Iran, 2010.

[14] R. Roy, P. Bhatt, S.P. Ghoshal, "Evolutionary computation based three-area automatic generation control", Expert Systems with Applications, Article in Press (2010).

[15] O. Bozorg Haddad, A. Afshar, M. A. Mariño, Honey-bees mating optimization (HBMO) algorithm: a new heuristic approach for water resources optimization", Water Resources Managament, Vol. 20, pp. 661-680, 2006.

[16] F. Mohammad, A. Babak, M. Ali, "Application of honey-bee mating optimization algorithm on clustering", Applied Mathematics and Computations, Vol., 1502-1513, 2007.

[17] H. Shayeghi, H. A. Shayanfar, O. P. Malik, "Robust decentralized neural networks based LFC in a deregulated power system", *Electric Power Systems Research*, Vol. 77, pp. 241-251, 2007.

[18] H. Shayeghi, H. A. Shayanfar, O. P. Malik, "Robust decentralized neural networks based LFC in a deregulated power system", *Electric Power Systems Research*, Vol. 47, pp. 241-251, 2007.

[19] B. Amiri, M. Fathian, "Integration of self organizing feature maps and honey bee mating optimization algorithm for market segmentation", Journal of Theoretical and Applied Information Technology, pp. 70-86, 2007.

[20] M. Fathian, B. Amiri, "A honeybee-mating approach for cluster analysis", International Journal of Advance Manufacturing and Technology, Vol. 38, pp. 809-821, 2008.

[21] H. Shayeghi, A. Jalili, H. A. Shayanfar, "Multi-stage fuzzy load frequency control using PSO", Energy Conversion and Management, Vol. 49, pp. 2570-2580, 2008.

Development of Discriminant Analysis and Majority-Voting Based Credit Risk Assessment Classifier

Paulius Danenas[1], **Gintautas Garsva**[2], **and Rimvydas Simutis**[3]
[1,2,3] Department of Informatics, Kaunas Faculty of Humanities, Vilnius University, Kaunas, Lithuania

Abstract - *This article presents a research on a method for credit risk evaluation combining expert majority-based ensemble voting scheme together with discriminant analysis as basis for expert formation and popular machine learning techniques for classification, such as decision trees, rule-based inducers and neural networks. Both single expert and multiple expert evaluations were applied as basis for forming output classes dynamically. Feature selection was applied using correlation-based feature subset evaluator with tabu search. The experiment results form a basis for further research of similar method.*

Keywords: machine learning, credit risk, bankruptcy, majority voting, discriminant analysis, decision trees

1 Introduction

Credit risk evaluation is a widely researched problem which is very important to banks, credit unions and other financial institutions which have to evaluate the possibility of default and to decide whether to satisfy credit request or to reject it. It is especially important in periods of financial crisis as many companies are going bankrupt and this task becomes even more complicated. There are many factors to be evaluated, including current macroeconomic situation, financial ratios that represent the customer's current situation and his financial history. A lot of statistical, econometric, mathematical methods are applied and used in this field, as well as methods based on expert knowledge, artificial intelligence, machine learning. The most widely used techniques are based on scoring and evaluation of probability default. However, many researchers currently focus on intelligent techniques to solve classification and forecasting tasks related to this filed, such as bankruptcy prediction, client evaluation, risk assessment. These methods include neural networks, rule-based classifiers, decision trees, state-of-the-art techniques such as Support Vector Machines (SVM), also modern heuristic techniques such as evolutionary computing or swarm intelligence. These classifiers are applied as standalone and in ensemble metaclassifiers which often obtain better predictive performance and produce better results. Majority voting can also be used while trying to find an optimal solution if more than one expert gives his evaluation on particular problem; it takes the decision my the biggest number of votes. This research proposes a model for credit risk evaluation which combines machine learning classification procedures and majority-voting based expert evaluation principles. It can be used in cases when there are several predictions for each instance.

2 Related work

Many of the earliest researches were based on discriminant analysis. The most widely known and used was developed in 1968 by Altman et al. [1]. Altman obtained 96% and 79% accuracy by using two different samples, however, it is reliable in its predictive ability only in two years, after that the results fall down significantly. Zmijewski [2] examined two estimation biases for financial distress models on non-random samples by using probit (simple probit and bivariate) and maximum likelihood principles. His data set consisted of estimation sample of 40 bankrupt and 800 non-bankrupt companies and a prediction sample of 41 bankrupt and 800 non-bankrupt companies collected from American and New York Stock Exchanges. Springate [3] developed his model using step-wise multiple discriminate analysis to select 4 ratios which best describe a failing company. It obtained an accuracy rate of 92.5% using the 40 companies tested by Springate; later 83.3% and 88% accuracy rates were reported after testing it with other samples [4]. Shumway [5] forecasted bankruptcies with market-driven variables exclusively and by combining market-driven variables with two accounting ratios from Zmijewski's model with data (each firm-year) from 1962 to 1992 (33,621 firm-years and 291 bankruptcies in the first case, 28,664 firm-years and 239 bankruptcies in second case). In the first case 69 percent of bankrupt firms were in the highest probability decile and 95 percent of bankrupt firms above the probability median.

Neural networks have been used for research in credit evaluation field since they were applied as computational technique. Such researches include learning vector quantization (LVQ) network [6], fuzzy neural networks with particle swarm optimization for parameter selection [7], evolutionary neural networks [8] and many other. Self-organizing maps, often referred as Kohonen maps, were also successfully applied [9][10]. Support Vector Machines (SVM) has been extensively researched recently in this field and has been proved to be very efficient obtaining results that can be compared to Neural Networks. Danenas et. al applied LIBLINEAR and SMO algorithms [11], combined with

discriminant analysis for evaluation, achieving results similar to Vapnik's SVM classifier results. SVM method has been combined with almost all popular natural computing techniques while applying it in credit risk assessment and bankruptcy prediction; many of these investigations related to credit risk evaluation using SVM-based methods are discussed in [12]

Decision tree is one of the oldest and most widely applied machine learning techniques. One can find numerous applications in various fields, including finance. It comprises a large family of algorithms which were developed on its basis - Classification and Regression Trees (CART), Chi-squared Automatic Interaction Detection (hence CHAID), C4.5 by Quinlan [13]. Modern techniques include functional trees (FT) with logistic regression functions at the inner nodes [14] and logistic model trees (LMT) [15], combination of DT and Naïve Bayes with NB classifiers at leaves [16] as well as forests of random trees [17]. These algorithms tend to show promising results thus they will be used in our experiment as classification methods.

3 Research method

3.1 Binary majority voting evaluation

The "expert" majority evaluation algorithm is based on majority voting principles used in ensemble classifiers, although it has some major changes. This algorithm can be expressed as follows in Figure 1.

Input: *m* – number of "experts" (uncorrelated evaluators), *C* – set representing possible class values ($C \in N$ and $C = N_0 \backslash N_c$, as we analyse only binary classification here, $C = \{0,1\}$), *M* - predictions of experts with values from set *C*, M_j - prediction of *j*-th "expert" such that $Mj \in C$, $j = 1..m$.

1. if (m = 1)
2. $y = M_1$ *(we have single output, nothing to be done)*
3. else-if (m = 2)
4. if ($M_1 \neq M_2$)
5. $y = \arg\max_{c \in M} \sum_{i:M_i=c} 1$ *(simple majority selection)*
6. else
7. $y = rand(\sum_{i:M_i=c} 1)$ *(select value by random)*
8. else-if (*m = 2n-1* and *n ≥2*)
9. $y = \arg\max_{c \in M} \sum_{i:M_i=c} 1$ *(simple majority selection)*
10. else-if (*m = 2n* and *n ≥2*) {
11. k_0= size({i:M_i=c_0})
12. k_1= size({i:M_i=c_1})
13. if ($k_0 \neq k_1$)
14. $y = \arg\max_{c \in M} \sum_{i:M_i=c} 1$ *(simple majority selection)*
15. else {
16. Θ = {} *(init an empty set of "expert" groups)*
17. For k=1 to m do {
18. M´ = rem(M, k) *(remove k-th element from M)*
19. Θ = add(Θ, M´) *(add formed group to set of experts)*
20. }
21. (*remove one ensemble from set by random)*
22. Θ=remove(Θ, rand(1, m))
23. $y = \arg\max_{c \in \Theta} \sum_{i:\Theta_i=c, e \in \Theta} \arg\max_{c' \in e} \sum_{j:e_i=c'} 1$
24. }
25. }

Output: *y* - output value for instance D_i of dataset *D*.

Figure 1. Pseudocode for binary majority voting evaluation algorithm

A more detailed explanation of algorithm for case when $m \in \{2n; \forall n \in N; \forall n \geq 2\}$ is as follows: if simple majority evaluation is not possible, we create *m* ensembles (groups of "experts") with *m-1 = 2n-1* members (such that we can apply simple majority voting principle) and randomly select *m-1 = 2n-1* evaluations from here such that expert would participate in this evaluation at least *m-1* times. Thus group majority voting evaluation is decomposed into a set of decisions by subgroups and the evaluation is obtained voting these decisions.

However, it becomes a difficult task to decide which evaluation should be selected if m = 2 and $M_1 = M_2$ as we have two different evaluations and no voting can be applied. Random selection was chosen to in this experiment to solve this problem; however, other options might be application of weights for each of "experts". If evaluators are other classifiers, it might be appropriate to select their accuracy or other evaluation metrics.

3.2 Proposed method

This section describes a method based on genetic search, machine learning technique for classification and discriminant analysis. The main steps are as follows:

1. Evaluate every instance by using *k* evaluators. In this experiment these "experts" are based on discriminant analysis models which values are converted to bankruptcy classes. If there are instances with empty outputs (records, which couldn't be evaluated in Step 1 because of lack of data or division by zero), the evaluation is marked as N/A and is excluded from instance evaluation. If all k evaluations are marked as "N/A", the instance is eliminated.

2. Data preprocessing:

a. Data imputation to eliminate empty values; here missing values are replaced with company's average of the attribute with missing data (for dataset *D* with attributes *X* and its subset D_C as financial records (instances) related to company C, if $D_{Cij} = \{\}$ then $D_{Cij} = average(X_i)$, $i=1..,m$, $j=1,..,n$; here *m* is the number of attributes, *n* – length of D_C);

b. Create training and testing data by splitting data of selected companies in the sector by particular percentage for hold-out training. These sets are disjoint (for dataset $D = D\_train \cup D\_test$, and $|D\_train| > |D\_test|$);

3. Apply feature selection to select the most relevant ratios;

4. Train classifier using one of machine learning classification algorithms;

5. The created model is tested using testing (holdout) data and results are evaluated.

3.3 Methods for evaluation of instances and results

Discriminant analysis was selected as basis for expert evaluation as widely applied and cited method.

Table I. Discriminant models used in evaluation

	Altman (original)	**Springate**	**Zmijewski**	**Shumway**
w_0	-	-	-4.336	-7.811
w_1	1.2	1.03	-4.513	-6.307
x_1	Working capital/Total assets	Working capital/Total assets	Net Income/Total assets	Net Income/Total assets
w_2	1.4	3.07	5.679	4.068
x_2	Retained earnings/Total assets	Net Profit before Interest and Taxes/Total Assets	Total liabilities/Total assets	Total liabilities/Total assets
w_3	3.3	0.66	0.004	-0.158
x_3	Earnings before interest and taxes/ Total assets	Net Profit before Taxes/Current Liabilities	Current assets / current liabilities	Current assets / current liabilities
w_4	0.6	0.4	-	-
x_4	Book value of Equity/ Book value of total liabilities	Sales/Total Assets	-	-
w_5	0.999	-	-	-
x_5	Net sales/Total assets	-	-	-
Eval	Z>3 – healthy; 2.7<Z<2.99 – non-bankrupt; Z <1.79 - bankrupt	Z>0.862 – healthy, Z<0.862 - bankrupt	Z>0– healthy, Z<0 - bankrupt	Z>0– healthy, Z<0 - bankrupt

Four models used in USA and Canada (as data used in experiment is from EDGAR database), particularly Altman Z-Score, Springate, Zmijewski and Shumway (essentially hazard modification of Zmijewski model) were applied; they are listed in Table I.

Algorithms described in section 3.1 were used in this experiment to train models. The test results are evaluated by using accuracy together with TP (True Positive) and F-Measure rates. As most of the experiment is concluded for two class only (except for single "expert" based on Altman) we give their definitions in terms of binary classification. Accuracy is defined as a proportion of correct predictions to total predictions as

$$acc = \frac{TP + TN}{TP + FP + FN + TN} \quad (1)$$

True Positive rate (also referred as Recall rate) is evaluated as a ratio of true predictions and number of total positive instances:

$$TPR = \frac{TP}{TP + FN} \quad (2)$$

Precision is a rate of predicted positive cases and total number of positive predictions (true positives and false positives):

$$prec = \frac{TP}{TP + FP} \quad (3)$$

F-Measure can be defined as a better option for evaluation of classifier trained with unbalanced data than accuracy and is defined as harmonic mean of precision and recall:

$$F_1 = \frac{2 * prec * recall}{prec + recall} \quad (4)$$

4 The experiment

4.1 Research data

The algorithm described in Section 3.2 was applied on a dataset consisting of entries from 1354 USA service companies with their 2005-2007 yearly financial records (balance and income statement) from financial EDGAR database. Each instance has 62 financial attributes (indices used in financial analysis).

Two types of datasets were formed: three consisting of single "expert" evaluations, and three consisting of majority-based evaluation described in section 3.2. To level the number of classes formed by discriminant analysis based "experts" it was reduced to the smallest number of classes provided by an expert; thus Altman-based "expert" formed only classes "bankrupt" and "good" instead of three own classes by its

original evaluation by combining "average" and "healthy" classes. The main statistics of these datasets is given in Table II.

Table 2. Dataset statistics

Dataset	Instances	No of ratios	No classes
Altman	3321	13	3
Springate	3245	12	2
Zmijewski	3245	12	2
Springate-Zmijewski-Shumway	3245	9	2
Altman-Springate-Zmijewski-Shumway	3369	10	2
Altman-Springate-Zmijewski	3372	18	2

Figure 1 gives an overview of evaluation statistics (distribution of classes that is produced by each expert) for the case of all four „experts". This distribution is clearly unbalanced; this is the case of application in real world when a small number of instances labelled as „bankrupt" might be sampled together with much larger number of instances labelled as „average" or „healthy".

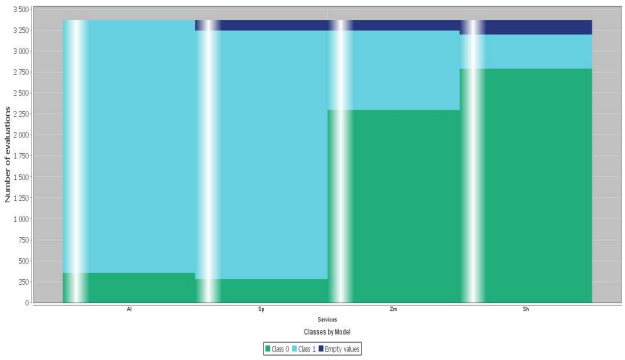


Figure 2. Expert evaluation statistics

There are three classes: Class 0 (mapped to „bankrupt" class), Class 1 (mapped to „healthy" class) and „Empty values" (or unevaluated, marked as "N/A"); the latter one presents the classes that could not be evaluated (e.g., because of lack of particular data or division by zero). This distribution gives just an overview of evaluation; as it represents the distribution after the instances that could not be evaluated (i.e., all evaluations were marked as „N/A") were removed, it might slightly differ for each case. Main properties of the datasets, together with number of removed (marked as „N/A") instances, are presented in Table II. It also shows a big difference between each "expert" class distribution, i.e., it can be viewed as two groups each consisting of two experts. This contradiction results in a smaller number of total "agreements" in cases when one particular class totally dominates, and makes it more sensible to explore this principle. However, the case of one class domination might mean that there is an agreement, and thus there is a higher chance that group decision is correct.

4.2 Experiment configuration

The experiment was executed using implementations of neural networks (RBF Network, Multilayer Perceptron), rule based classifiers JRip and FURIA (Fuzzy Unordered Rule Induction Algorithm) and decision tree based algorithms best-first decision tree (BFTree), FT, C4.5, LADTree, LMT, NBTree, RandomForest, REPTree (Weka's implementation of fast decision tree learner) and SimpleCart (minimal cost-complexity pruning DT) as classifiers. Parameters suggested by default were selected for each of these classifiers. To improve results, Bagging procedure was applied to each of them. All these classifiers are implemented in Weka machine learning framework.

A 7:3 split (70% percent of instances were selected for training) was used in the experiment. Feature selection was applied for each dataset using correlation-based feature subset evaluator with Tabu search. This resulted in reduced number of dimensions and less complexity. Three groups of experts were formed for evaluation: Altman-Springate-Zmijewski-Shumway (further referred as Al-Sp-Zm-Sh), Altman-Springate-Zmijewski (further - Al-Sp-Zm) and Springate-Zmijewski-Shumway (further - Sp-Zm-Sh). Single expert evaluations (Altman, Springate, Zmijewski) were also applied for model training. However, two experts' evaluation based models were not chosen for evaluation as they might result in too many random evaluations and the model would be inconsistent; thus this selection is left only for evaluation in case of missing values.

4.3 Experiment results

The results of these experiments are presented in Table 3 and Table 4, which include TP rates together with F-Measure ratios. F-Measure is a good option for evaluation as it offers a trade-off between precision and recall (or TP rate). Table 3 presents the results of single "expert" based classifiers. Rule based classifiers and tree-based classifiers outperformed neural network based classifiers.

Result analysis shows that NN-based classifiers performed poorly while identifying particular classes (such as "bankrupt" and "average" in case of Altman-based evaluation) thus they cannot deal with dataset imbalance and special methods for dealing with imbalanced data, such as sampling-based (undersampling, oversampling) or cost-sensitive learning, should be applied together with these methods. The results obtained by decision tree and rule-based classifiers were better and they also identified bankrupt classes more precisely.

Table 4 presents results of majority-based evaluation of multiple expert classifier results. The results show that rule based and tree-based classifiers outperformed neural networks. Both these types of classifiers are better for unbalanced type of data; the results in Table 4 prove this, as true predictions ratio of both "bankrupt" and "healthy" companies" is higher than neural-network based classifiers. "Bad" companies were identified almost perfectly by Sp-Zm-

Sh model; however, ensemble of all four experts performed poorly while identifying bankrupt companies. This might be a consequence of the fact that original Altman model consists of three classes and the combination of the last two as "healthy" class resulted in very high imbalance.

Table 3. Single expert based classifier result

			RBF Network	Multilayer NN	FURIA	JRip	BFTree	FT	C4.5	LAD Tree	LMT	NBTree	Random Forest	REP Tree	Simple Cart
Altman	Accuracy		83,94	82,63	87,75	86,35	85,54	86,75	86,15	84,64	86,15	86,55	87,45	86,75	85,24
	TP	B	0,43	0,32	0,68	0,59	0,58	0,67	0,58	0,56	0,63	0,59	0,64	0,62	0,55
		A	0,00	0,00	0,24	0,22	0,22	0,25	0,25	0,20	0,25	0,25	0,22	0,25	0,20
		G	0,98	0,98	0,97	0,97	0,96	0,96	0,96	0,95	0,95	0,96	0,97	0,96	0,96
	F-Measure	B	0,53	0,42	0,69	0,62	0,61	0,67	0,63	0,56	0,65	0,63	0,67	0,65	0,58
		A	0,00	0,00	0,33	0,30	0,26	0,34	0,31	0,25	0,30	0,35	0,30	0,33	0,25
		G	0,91	0,90	0,94	0,93	0,94	0,93	0,93	0,93	0,94	0,93	0,94	0,94	0,93
Springate	Accuracy		94,35	93,94	97,95	97,64	97,74	95,99	97,84	97,74	97,43	97,43	97,53	97,33	97,95
	TP	B	0,56	0,61	0,85	0,80	0,85	0,78	0,84	0,83	0,78	0,79	0,80	0,81	0,87
		G	0,98	0,97	0,99	0,99	0,99	0,98	0,99	0,99	0,99	0,99	0,99	0,99	0,99
	F-Measure	B	0,64	0,64	0,88	0,86	0,87	0,78	0,87	0,87	0,84	0,85	0,85	0,84	0,88
		G	0,97	0,97	0,99	0,99	0,99	0,98	0,99	0,99	0,99	0,99	0,99	0,99	0,99
Zmijewski	Accuracy		93,83	83,04	97,95	98,15	96,51	99,38	96,51	96,81	99,49	97,23	98,36	97,43	96,40
	TP	B	0,99	0,99	0,99	0,99	0,98	1,00	0,98	0,98	1,00	0,98	0,99	0,98	0,99
		G	0,84	0,49	0,97	0,96	0,93	0,99	0,94	0,94	0,99	0,96	0,98	0,96	0,92
	F-Measure	B	0,96	0,89	0,99	0,99	0,97	1,00	0,97	0,98	1,00	0,98	0,99	0,98	0,97
		G	0,90	0,65	0,97	0,97	0,95	0,99	0,95	0,95	0,99	0,96	0,98	0,96	0,94

However, inclusion of Altman-based expert also did not result in high value of TP for "bankrupt" class in Al-Sp-Zm model which proves that Altman based "expert" was the one that imbalanced the results.

Table 4 shows that best results were obtained by Sp-Zm-Sh and Al-Sp-Zm evaluation based classifiers. It is not surprising as these models used simple majority voting (i.e., there was mostly majority-based consensus, as Figure 1 shows), without the need to form additional inner ensembles. However, the results show that Sp-Zm-Sh based classifier obtained TP for instances labeled as "bankrupt" values close to 1, to compare with values ranging from 0.78-0.85 in case of Springate model applied alone for evaluation. This increase is even higher in case of neural network based classifiers (from 0.56 and 0.61 to 0.98 and 0.96 respectively). However, TP rate values for instances labeled as "good" became lower, especially in case of NN based classifiers. Yet this value did not decrease much in cases of rule-based and tree-based classifiers where it still remained above 0.9. This concludes that multiple "expert" evaluation not only increased reliability of single "expert" based models, but it also sort-of helped to overcome imbalance barrier which NN classifiers seem to have.

Table 4. Single expert based classifier result

	Springate-Zmijewski-Shumway					Altman-Springate-Zmijewski-Shumway					Altman-Springate-Zmijewski				
	Acc	TP		F-Measure		Acc	TP		F-Measure		Acc.	TP		F-Measure	
		B	G	B	G		B	G	B	G		B	G	B	G
RBF Network	88,59	0,98	0,68	0,92	0,79	78,46	0,07	0,99	0,13	0,88	93,38	0,39	0,99	0,51	0,96
Multilayer NN	86,33	0,96	0,66	0,91	0,75	77,67	0,05	0,99	0,10	0,87	91,90	0,41	0,97	0,47	0,96
FURIA	96,40	0,97	0,94	0,97	0,94	78,85	0,18	0,97	0,28	0,88	94,37	0,56	0,98	0,64	0,97
JRip	96,81	0,97	0,96	0,98	0,95	78,56	0,15	0,97	0,23	0,88	94,27	0,54	0,98	0,63	0,97
BFTree	94,76	0,97	0,90	0,96	0,92	78,56	0,16	0,97	0,25	0,88	94,47	0,57	0,98	0,65	0,97
FT	97,02	0,98	0,95	0,98	0,95	79,55	0,22	0,96	0,32	0,88	93,97	0,56	0,98	0,62	0,97
C4.5	95,48	0,98	0,91	0,97	0,93	78,46	0,20	0,96	0,29	0,87	94,37	0,57	0,98	0,64	0,97
LADTree	94,55	0,96	0,91	0,96	0,91	78,76	0,19	0,96	0,29	0,88	94,66	0,57	0,98	0,65	0,97
LMT	96,71	0,97	0,96	0,98	0,95	78,26	0,20	0,95	0,30	0,87	94,07	0,59	0,98	0,64	0,97
NBTree	95,79	0,97	0,93	0,97	0,93	79,55	0,20	0,97	0,30	0,88	94,07	0,48	0,99	0,59	0,97
RandomForest	97,33	0,98	0,97	0,98	0,96	78,85	0,20	0,96	0,30	0,88	95,16	0,59	0,99	0,68	0,97
REPTree	96,10	0,97	0,95	0,97	0,94	78,56	0,22	0,95	0,32	0,87	93,48	0,50	0,98	0,58	0,97
SimpleCart	95,38	0,97	0,92	0,97	0,93	77,37	0,19	0,94	0,28	0,87	94,76	0,58	0,98	0,66	0,97

However, inclusion of Altman evaluation based results resulted in significantly different results. Both evaluation models that were tested (Altman-Springate-Zmijewski-Shumway and Altman-Springate-Zmijewski) obtained results

which are worse in terms of TP values that corresponding single “expert” based models. Analysis of TP results for instances labeled as “bankrupt” shows that the most significant influence was the imbalance of data (instances both labeled as “average” and “good” were combined as “good”). Although the Al-Sp-Zm evaluation based classifier results did not suffer much, Al-Sp-Zm-Sh performed poorly in terms of TP rate for “bankrupt” classes. A more concise Altman-based model (i.e., having more classes, which might help to make a better balancing while “splitting” into binary categories) or proper dataset balance might help to overcome this problem.

5 Conclusions and further research

This article presents a research on credit risk which combines machine learning classifiers and multiple majority-voting based “expert” evaluation by discriminant analysis. This method can be complemented by feature selection or extraction procedure; this research used correlation-based feature selection. However, this method currently is only suitable for binary classification; it might be extended for multiclass evaluation in the future. Another important issue is learning from unbalanced data; if modern techniques such as neural networks or SVM are selected as classifiers, techniques to overcome this barrier should be applied, such as internally implemented class-weighting, cost-sensitive learning and evaluation, internal classifier enhancements or sampling techniques. The experiment provided promising results, especially when identifying “bankrupt” companies. Further research might be targeted at multiclass extension, integration of other soft computing techniques such as fuzzy logic or rough sets, as well as optimization of various classifiers by best parameter selection.

6 References

[1] Edward Altman. "Financial ratios, discriminant analysis and the prediction of corporate bankruptcy"; The Journal of Finance, Vol. 23, pp. 589–609, 1968.

[2] Mark Zmijewski. "Methodological Issues Related to the Estimation of Financial Distress Prediction Models"; Journal of Accounting Research 22 (Supplement), pp. 59-86, 1984.

[3] Gordon L. V. Springate. “Predicting the Possibility of Failure in a Canadian Firm”; Unpublished M.B.A. Research Project, Simon Fraser University, 1978.

[4] Earl G. Sands, Gordon L.V. Springate, Turgut Var. "Predicting Business Failures"; CGA Magazine, pp. 24-27, 1983.

[5] Tyler Shumway. “Forecasting Bankruptcy More Accurately: A Simple Hazard Model”; Journal of Business, Vol. 74(1), pp.101-125, 2001.

[6] Yunna Wu, Zhaomin Si. "Application of RBF Neural Network Based on Ant Colony Algorithm in Credit Risk Evaluation of Construction Enterprises"; Proceedings of the 2008 International Conference on Risk Management \& Engineering Management (ICRMEM '08). IEEE Computer Society, Washington, DC, USA, pp.653-658, 2008.

[7] Fu-yuan Huang. "A Particle Swarm Optimized Fuzzy Neural Network for Credit Risk Evaluation"; WGEC '08 Proceedings of the 2008 Second International Conference on Genetic and Evolutionary Computing, pp. 153-157, 2008.

[8] Estefane Lacerda, André C. P. L. F. Carvalho, Antônio Pádua Braga Teresa Bernarda Ludermir, "Evolutionary Radial Basis Functions for Credit Assessment"; Applied Intelligence, Vol. 22, Number 3, pp. 167-181, 2005.

[9] Egidijus Merkevicius, Gintautas Garsva, Rimvydas Simutis. "Neuro-discriminate Model for the Forecasting of Changes of Companies Financial Standings on the Basis of Self-organizing Maps"; Lecture Notes In Computer Science, Vol. 4488, Berlin, Heidelberg: Springer-Verlag, 2007, pp. 439-446.

[10] Egidijus Merkevicius, Gintautas Garsva, Stasys Girdzijauskas. "A hybrid SOM-Altman model for bankruptcy prediction"; Lecture Notes in Computer Science, Vol. 3994, 2006, pp. 364-371.

[11] Paulius Danenas, Gintautas Garsva. “Credit risk evaluation using SVM-based classifier”; Business Information Systems Workshops: Lecture Notes in Business Information Processing, Vol. 57, Part 1, 2010, pp. 7-12.

[12] Paulius Danenas, Gintautas Garsva. “Support Vector Machines and their Application in Credit Risk Evaluation”; Transformations in Business & Economics, Vol. 8, No. 3 (18), 2009, pp. 46-58

[13] Ross Quinlan. C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[14] João Gama. “Functional trees”; Machine Learning, Vol. 55, No. 3, pp. 219-250, 2004.

[15] Niels Landwehr, Mark Hall, Eibe Frank. “Logistic Model Trees”; Machine Learning, Vol. 59, Issue 1-2, pp. 161-205, May 2005.

[16] R. Kohavi. ”Scaling Up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid”; Second International Conference on Knowledge Discovery and Data Mining, pp. 202-207, 1996.

[17] Leo Breiman. “Random Forests”; Machine Learning, Vol. 45(1), pp. 5-32, 2001.

PSS Design for a Single-Machine Power System Using Honey Bee Mating Optimization

H. Shayeghi
Technical Eng. Department
University of Mohaghegh Ardabili
Ardabil, Iran

H. A. Shayanfar
E.E.Department
Center of Excellence for Power System Automation and Operation
Iran University of Science and Technology, Tehran, Iran

A. Akbarimajd
Technical Eng. Department
University of Mohaghegh Ardabili
Ardabil, Iran

A. Ghasemi
Technical Eng. Department
University of Mohaghegh Ardabili
Ardabil, Iran

hshayeghi@gmail.com, hashayanfar@yahoo.com, akbarymajd@yahoo.com, ghasemi.agm@gmail.com

Abstract- *A Honey Bee Mating Optimization (HBMO) technique is proposed for the optimal tuning of the Power System Stabilizer (PSS) in this paper. The design problem of robustly selecting PSS parameters is formulated as an optimization problem according to the time domain-based objective function which is solved by the HBMO technique that has a strong ability to find the most optimistic results. To achieve desired level of robust performance and improve power system low frequency oscillations damping, The design process takes a wide range of operating conditions. The effectiveness of the proposed HBMO based PSS is demonstrated on a Single-Machine Infinite-Bus (SMIB) power system through the nonlinear time domain simulation and some performance indices under different operating conditions in comparison with the genetic algorithm based tuned stabilizer and conventional PSS. Results evaluation show that the designed HBMO based PSS has an excellent capability in damping power system low frequency oscillations and enhance greatly the dynamic stability of the power systems.*

Keywords: PSS Design, HBMO, Low Frequency Oscillations, SMIB.

1. Introduction

Nowdays, many power-generating plants were equipped with continuously acting Automatic Voltage Regulators (AVRs). As the number of power plants with AVRs grew, it became apparent that the high performance of these voltage regulators had a destabilizing effect on the power system. On the other hand, by the development of interconnection of large electric power systems, there have been spontaneous system oscillations at very low frequencies in order of 0.2-3.0 Hz. In some cases, they continue to grow and presented a limitation on the amount of the power to be transmitted within the system, if no adequate damping is available [1]. Power System Stabilizers (PSSs) are auxiliary control devices on synchronous generators, used in conjunction with their excitation systems to provide control signals toward enhancing the system damping of low frequency oscillations associated with the electromechanical modes and extend power transfer limits; thus maintaining reliable operation of the power system [2].

The conventional lead-lag compensator based stabilizers is still widely used for practical applications and generally provide acceptable dynamic performance. This is because it performs well for a wide class of process. Also, they give robust performance for a wide range of operating conditions and easy to implement. However, the problem of PSS parameter tuning is a complex exercise. A number of the conventional techniques have been reported in the literature pertaining to design PSS namely: the eigenvalue assignment, mathematical programming, gradient procedure for optimization and also the modern control theory [2-5]. The main problem encountered in the conventional PSS design is that the power system constantly experience changes in the operating conditions due to variation in generation and load patterns, as well as change in transmission network. Thus, the investigations carried out using these approaches reveal that it exhibits poor dynamics performance, especially in the presence of the other destabilizing effects such as parameter variations and nonlinearities. A more reasonable design of the PSS is based on the gain scheduling and adaptive control theory as it takes into consideration the nonlinear and stochastic characteristics of the power systems [6-7]. This type of stabilizer can adjust its parameters on-line according to the operating condition. Many years of intensive studies have shown that the adaptive stabilizer

* Corresponding Author. E-Mail Address: hashayanfar@yahoo.com (H. A. Shayanfar)

can not only provide good damping over a wide operating range but more importantly, it can also solve the coordination problem among the stabilizers. Many random heuristic methods, such as like Tabu search, genetic algorithms, chaotic optimization algorithm, rule based bacteria foraging and Particle Sswarm Optimization (PSO) have recently received much interest for achieving high efficiency and search global optimal solution in the problem space and they have been applied to the problem of PSS design [8-11]. These evolutionary based methods are heuristic population-based search procedures that incorporate random variation and selection operators. Although, these methods seem to be good approaches for the solution of the PSS parameter optimization problem, however, when the system has a highly epistatic objective function (i.e. where parameters being optimized are highly correlated), and number of parameters to be optimized is large, then they have degraded effectiveness to obtain the global optimum solution. In order to overcome these drawbacks, a Honey Bee Mating Optimization (HBMO) technique is proposed for the optimal tune of the PSS parameters to improve power system low frequency oscillations damping in this paper. The HBMO algorithm is a typical swarm-based approach to optimization, in which the search algorithm is inspired by the honey-bee mating process [12] and has emerged as a useful tool for the engineering optimization. It is a hybrid evolutionary algorithm which is comprised of GA, SA, local search, and some innovations for its self-adaptation. Unlike the other heuristic techniques, The HBMO algorithm has a flexible and well-balanced mechanism to enhance the global and local exploration abilities [13].

In this study, the problem of robust PSS design is formulated as an optimization problem and HBMO technique is used to solve it. A performance index is defined based on the system dynamics after an impulse disturbance alternately occurs in the system and used to form the objective function of the design problem. The proposed HBMO based designed PSS has been applied and tested on a weakly connected power system under wide range of operating conditions to illustrate their ability to provide efficient damping of low frequency oscillations. To show the superiority of the proposed design approach, the simulations results are compared with the GA based designed and classical PSS under different operating conditions through some performance indices. The results evaluation shows that the proposed method achieves good robust performance for wide range of load changes in the presence of very highly disturbance and is superior to the other stabilizers.

2. Power System Model

For the stability analysis of power system, adequate mathematical models describing the system are needed. The models must be computationally efficient and be able to represent the essential dynamics of the power system. The stability analysis of the system is generally attempted using mathematical models involving a set of nonlinear differential equations. A schematic diagram for the test system is shown in Fig. 1. The generator is equipped with the excitation system and a power system stabilizer. System data are given in the Appendix.

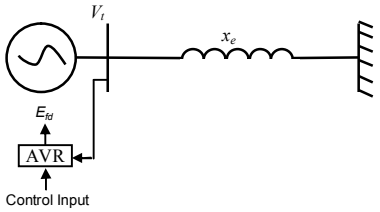


Fig. 1. SMIB power system

The synchronous generator is represented by model 1.1, i.e. with field circuit and one equivalent damper winding on q axis. The nonlinear dynamic equations of the SMIB system considered can be summarized as [14, 11].

$$\dot{\delta} = \omega_B S_m$$

$$\frac{dS_m}{dt} = \frac{1}{2H}(-DS_m + T_m - T_e)$$

$$\dot{E}'_q = \frac{1}{T'_{do}}(E_{fd} + (x_d - x'_d)i_d - E'_q) \quad (1)$$

$$\dot{E}_{fd} = \frac{1}{T_A}(k_A(v_{ref} - v_t + V_s)) - E_{fd})$$

$$T_e = E'_q i_q + (x'_d - x'_q) i_d i_q \quad (2)$$

2.1. PSS Model

The structure of the PSS, to modulate the excitation voltage is shown in Fig .2. The structure consists a gain block with gain K, a signal washout block and two-stage phase compensation blocks. The input signal of the proposed method is the speed deviation ($\Delta\omega$) and the output is the stabilizing signal V_S which is added to the reference excitation system voltage. The signal washout block serves as a high-pass filter, with the time constant T_W, high enough to allow signals associated with oscillations in input signal to pass unchanged. From the viewpoint of the washout function, the value of T_W is not critical and may be in the range of 1 to 20 seconds [11]. The phase compensation block (time constants T_1, T_2 and T_3, T_4) provides the appropriate phase-lead characteristics to compensate for the phase lag between input and the output signals.

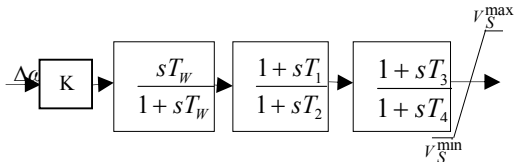


Fig .2. Structure of power system stabilizer

3. HBMO

A honey-bee colony typically consists of a single egg laying long-lived queen, several thousand drones (depending on the season), workers and a large family of bees living in one bee-hive [15]. Each bee undertakes sequences of actions which unfold according to genetic, ecological and social condition of the colony.

A mating flight starts with a dance performed by the queen who then starts a mating flight during which the drones follow the queen and mate with her in the air. After the mating process, the drones die. In each mating, sperm reaches the spermatheca and accumulates there to form the genetic pool of the colony. Each time a queen lays fertilized eggs, she randomly retrieves a mixture of the sperm accumulated in the spermatheca to fertilize the egg and this task can only be done by the queen [12-13].

The HBMO algorithm combines different phases of the marriage process of the honey bee. It starts with random generation of a set of initial solutions. Based on their fitness, randomly generated solutions are then ranked. The fittest solution is named queen, whereas the remaining solutions are categorized as drones (i.e., trial solutions). In order to form the hive and start mating process, the queen, drones and workers (predefined heuristic functions) should be defined. Each queen is characterized with a genotype, speed, energy and a spermatheca with defined capacity. In the next step, drones must be nominated to mate with the queen probabilistically during the mating flight. At the start of the flight, the queen is initialized with some energy content and returns to her nest when the energy is within some threshold of either near zero or when the spermatheca is full. The mating flight may be considered as a set of transitions in a state-space (the environment). An annealing function is used to describe the probability of a Drone (D) that successfully mates with the Queen (Q) as follows [12]:

$$prob(Q, D) = e^{\frac{-\Delta(f)}{S(t)}} \quad (3)$$

Where, $\Delta(f)$ is the absolute difference of the fitness of D and the fitness of Q and the $s(t)$ is the speed of queen at time t. The fitness of the resulting chromosomes of drone, queen or brood is determined by evaluating the value of the objective function. After each transition in space, the queen's speed and energy decays is given by:

$$S(t+1) = \alpha \times S(t) \quad (4)$$

$$E(t+1) = E(t) - \gamma \quad (5)$$

Where, $\alpha(t)$ is the speed reduction factor and γ is the amount of energy reduction after each transition (α , $\gamma \in [0,1]$).

In order to develop the algorithm, the capability of workers is restrained in brood care and thus, each worker may be regarded as a heuristic that acts to improve and/or take care of a set of broods. The rate of improvement in the brood's genotype, defines the heuristic fitness value. The fitness of the resulting genotype is determined by evaluating the value of the objective function of the brood genotype and/or its normalized value. It is important to note that a brood has only one genotype [16].

In general, the whole process of HBMO algorithm as can be summarized at the five main steps as follows:

i) Generate the initial drone sets and queen: The algorithm starts with the mating flight, where a queen (best solution) selects drones probabilistically to form the spermatheca (list of drones). A drone then selected from the list randomly for the creation of the broods.

ii) Flight matting: This steps do the flight matting of queen Q. The best drone D_k with the largest *prob(Q, D)* among the drone set D is selected the object of matting for the queen Q. After the flight matting the queen's speed and energy decay is reduced by Eq. (4). The flight matting is continues until the speed $s(t)$ is less than a threshold d or the number of sperms of the queen's spermatheca is less than the one threshold.

iii) Breeding process: In this step, a population of broods is generated based on the matting between the queen and the drones stored in the queen's spermatheca. The breeding process can transfer the genes of drones and the queen to the j-th individual based on the Eq. (6).

$$child = parent1 + \beta(parent2 - parent1) \quad (6)$$

Where, β is the decreasing factor ($\beta \in [0,1]$).

iv) Adaptation of worker's fitness: The population of broods is improved by applying the mutation operators as follows:

$$Brood_i^k = Brood_i^k \pm (\delta + \varepsilon) Brood_i^k \quad (7)$$
$$\delta \in [0,1], 0 < \varepsilon < 1$$

The δ is randomly generated and ε is predefined. The best brood ($brood_{best}$) with maximum objective function value is selected as the candidate queen. If the objective function of the $brood_{best}$ is superior to the queen, the queen replace with $brood_{best}$.

v) Check the termination criteria: If the termination criteria satisfied finish the algorithm, else generate new drones set and go to step 2.

The algorithm continues with three user-defined parameters and one predefined parameter. The predefined parameter is the number of Workers (W), representing the number of heuristics encoded in the program [12, 15]. The user-defined parameters are number of queens, the queen's spermatheca size representing the maximum number of mating per queen in a single mating flight and the number of broods that will be born by all queens. The speed of each queen at

the start of each mating flight initialized randomly. Since this algorithm is combination of simulated

annealing, genetic operator and swarm intelligence, it is very interesting optimization algorithm that used in the optimization problems of reservoir operation.

4. Problem Formulation

In case of the above lead-lag structured PSS, the washout time constants is usually specified. In the present study, washout time constant T_W =10 sec is used. The controller gain K and the time constants T_1, T_2, T_3 and T_4 are to be determined. It is worth mentioning that the PSS is designed to minimize the power system oscillations after a large disturbance so as to improve the power system stability. These oscillations are reflected in the deviations in power angle, rotor speed and line power. Minimization of any one or all of the above deviations could be chosen as the objective. In this study, an Integral Square Time of Square Error (ISTSE) of the speed deviations is taken as the objective function expressed as:

$$J = \sum_{i=1}^{NP} \int_{t=0}^{t=tsim} t^2 (\Delta\omega)^2 dt \qquad (8)$$

Where, $\Delta\omega$ denotes the rotor speed deviation for a set of PSS parameters, t_{sim} is the time range of the simulation and NP is the total number of operating points for which the optimization is carried out. It is aimed to minimize this objective function in order to improve the system response in terms of the settling time and overshoots under different operating condition. The design problem can be formulated as the following constrained optimization problem, where the constraints are the controller parameters bounds [9, 11]:

Minimize J Subject to:

$$\begin{aligned} K^{\min} &\le K \le K^{\max} \\ T_1^{\min} &\le T_1 \le T_1^{\max} \\ T_2^{\min} &\le T_2 \le T_2^{\max} \\ T_3^{\min} &\le T_3 \le T_3^{\max} \\ T_4^{\min} &\le T_4 \le T_4^{\max} \end{aligned} \qquad (9)$$

Typical ranges of the optimized parameters are [0.01-50] for K and [0.01-1] for T_1, T_2, T_3 and T_4. The proposed approach employs HBMO to solve this optimization problem and search for an optimal or near optimal set of PSS parameters. The optimization of the PSS parameters is carried out by evaluating the objective cost function as given in Eq. (8), which considers a multiple of operating conditions are given in Table 1. The operating conditions are considered for wide range of output power at different power factors. Results of the PSS parameter set values based on the objective function J, by applying a three phase-to-ground fault for 100 ms at generator terminal at t=1 sec using the proposed HBMO and GA algorithms are given in Table 2. The Classical PSS (CPSS) is design using the tuning guidelines given in [14] for the nominal operating point. Fig. 3 shows the minimum fitness functions evaluating process.

Table 1. Operation conditions

Case No.	P	Q	x_e	H
Case1 (Base case)	0.8	0.4	0.3	3.25
Case 2	0.5	0.1	0.3	3.25
Case 3	1	0.5	0.3	3.25
Case 4	0.8	0.4	0.6	3.25
Case 5	0.5	0.1	0.6	3.25
Case 6	1	0.5	0.6	3.25
Case 7	0.8	0	0.6	3.25
Case 8	1	-0.2	0.3	3.25
Case 9	0.5	-0.2	0.6	3.25
Case 10	1	0.2	0.3	0.81

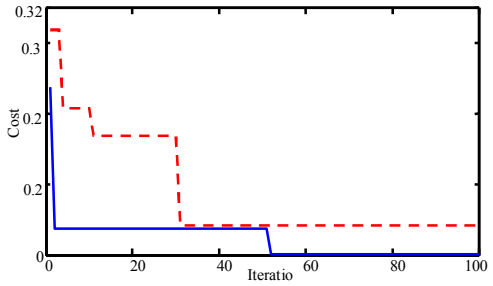


Fig. 3: Fitness convergence, Dashed (GA) and Solid (HBMO).

Table 2. Optimal PSS parameters

Algorithm	K	T_1	T_2	T_3	T_4
Classic	0.8595	0.38	2.278	2.0033	0.8526
GA	0.21134	0.09344	0.56014	0.49259	0.20965
HBMO	0.2756	0.7603	1.4967	0.706	0.5697

5. Simulation Results

The behavior of the proposed HBMO based designed PSS (HBMOPSS) under transient conditions is verified by applying disturbance and fault clearing sequence under different operating conditions. In comparison with the GA based tuned PSS (GAPSS) and classical PSS. The disturbances are given at t = 1 sec. System responses in the form of slip (S_m) are plotted. The following types of disturbances have been considered.

Scenario 1: A step change of 0.1 pu in the input mechanical torque.

Scenario 2: A three phase-to-ground fault for 100 ms at the generator terminal.

Figure 4 shows the system response at the lagging power factor operating conditions with weak transmission system for scenario 1. It can be seen that the system with CPSS is highly oscillatory. HBMO and GA based tuned stabilizers are able to damp the oscillations reasonably well and stabilize the system at all operating conditions. Figure 5 depicts the responses of same operating conditions but, with strong transmission system. System is more stable in this case, following any disturbance. Both PSSs improve its dynamic stability considerably and HBMOPSS shows its superiority over GAPSS and CPSS. Figure 6 refers to a three-phase to ground fault at the generator terminal. Figure 7 depicts the system response in scenario 1 with inertia H'= H/4. It can be seen that the proposed HBMO based PSS has good performance in damping low frequency oscillations and stabilizes the system quickly.

Moreover, it is superior to the GA and classical based methods tuned stabilizer.

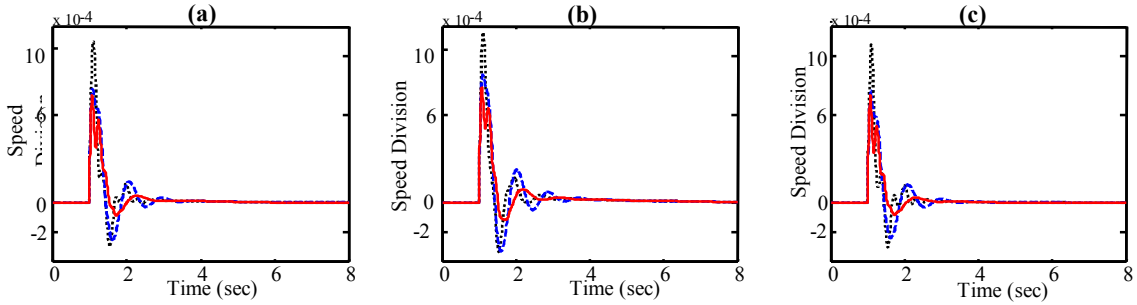


Fig. 4. ΔT_m=0.1 (p.u.) under X_e=0.3; CPSS (Dotted), PSOPSS (Dashed) and HBMOPSS (Solid)
a) P=0.8, Q=0.4 b) P=0.5, Q=0.1 c) P=1.0, Q=0.5

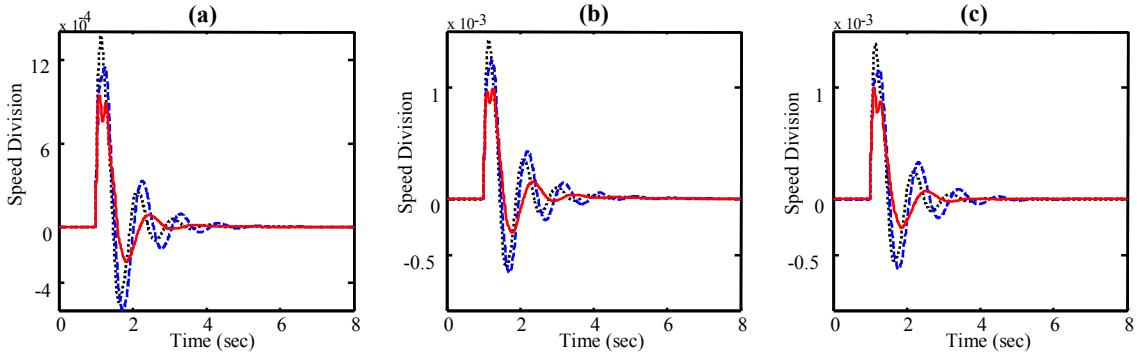


Fig. 5. ΔT_m=0.1 (p.u.) under X_e=0.6; CPSS (Dotted), GAPSS (Dashed) and HBMOPSS (Solid)
a) P=0.8, Q=0.4 b) P=0.5, Q=0.1 c) P=1.0, Q=0.5

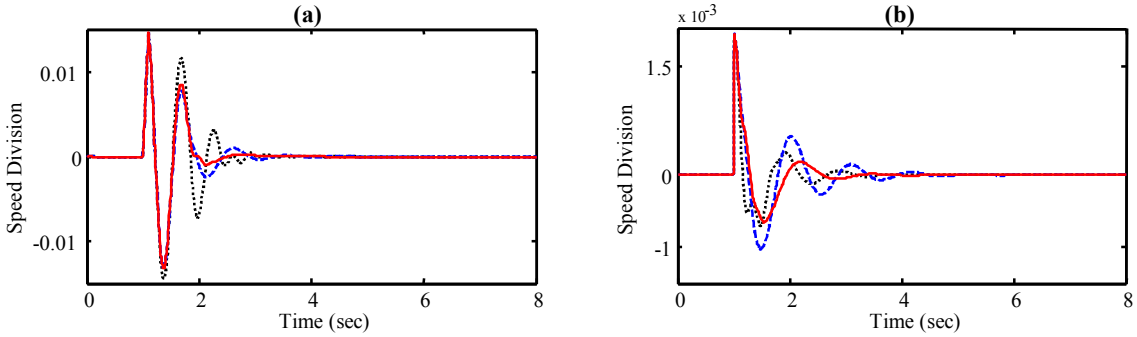


Fig. 6. 3-φ to ground fault 100 ms for X_e=0.3, CPSS (Dotted), GAPSS (Dashed) and HBMOPSS (Solid)
: a) P=0.8, Q=0.4 b) P=1.0, Q=0.5

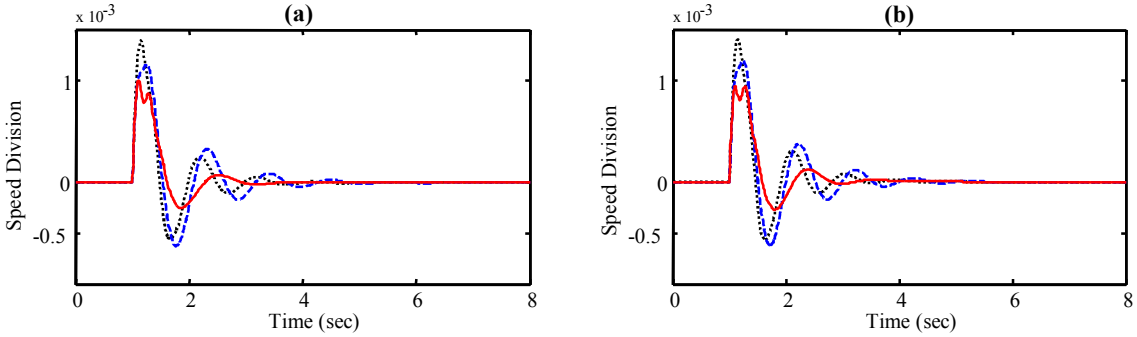


Fig. 7. ΔT_m=0.1 (p.u.) under X_e=0.6, CPSS (Dotted), GAPSS (Dashed) and HBMOPSS (Solid)
: a) P=1.0, Q=0.5 b) P=0.6, Q=0.0

To demonstrate performance robustness of the proposed method, two performance indices: the Integral of the Time multiplied Absolute value of the Error (ITAE) and Figure of Demerit (FD) based on the system performance characteristics are defined as [17]:

$$FD = 10 \times [(500 \times OS)^2 + (8000 \times US)^2 + 0.01 \times T_s^{\,2}] \quad (10)$$

$$ITAE = 1000 \times \int_0^8 t \,|\Delta\omega|dt \quad (11)$$

Where, Overshoot (OS), Undershoot (US) and settling time of rotor angle deviation of machine is considered for evaluation of the FD. It is worth mentioning that the lower the value of these indices are, the better the system response in terms of time-domain characteristics. Numerical results of performance robustness for all cases as given in Table 1 for scenario 1 and 2 are listed in Table 3 and 4, respectively.

Table 3. Performance indices for scenario 1

Case No	HBMOPSS		GAPSS		CPSS	
	ITAE	FD	ITAE	FD	ITAE	FD
1	0.5022	0.4757	0.6775	0.9942	0.5632	1.4729
2	0.8381	0.5644	1.0422	1.4799	0.9042	1.7696
3	0.3780	0.4722	0.5544	0.9281	0.4465	1.4774
4	0.7822	1.1298	1.4842	3.7696	1.1311	3.4747
5	1.1247	1.4463	1.8102	4.5268	1.4959	4.2209
6	0.7777	1.2171	1.5885	4.0670	1.1631	3.6312
7	0.7822	1.1298	1.4842	3.7696	1.1310	3.4747
8	0.3780	0.4722	0.5544	0.9281	0.4465	1.4774
9	1.1247	1.4463	1.8102	4.5268	1.4959	4.2209
10	0.3780	0.4722	0.5544	0.9281	0.4465	1.4774

Table 4. Performance indices for scenario 2

Case No	HBMOPSS		GAPSS		CPSS	
	ITAE	FD	ITAE	FD	ITAE	FD
1	0.3948	2.0346	0.4481	2.8143	0.5743	11.4092
2	0.2378	1.0007	0.3010	1.8424	0.3469	5.7607
3	0.5508	3.1000	0.5893	3.7712	0.7134	15.2602
4	0.7120	3.0457	1.0050	4.6696	0.6551	3.8234
5	0.4470	1.3547	0.7070	2.2499	0.4638	1.9659
6	0.9572	5.0157	1.4244	7.8263	0.8964	6.0103
7	0.7113	3.0370	1.0048	4.6696	0.6554	3.8250
8	0.5494	3.0962	0.5888	3.7694	0.7129	15.2359
9	0.4468	1.3547	0.7068	2.2499	0.4634	1.9598
10	0.5503	3.0981	0.5889	3.7694	0.7139	15.2846

It can be seen that the values of these system performance characteristics with the proposed HBMOPSS are much smaller compared to that GA and classical based designed PSS. This demonstrates that the overshoot, undershoot, settling time and speed deviations of machine is greatly reduced by applying the proposed HBMO based tuned PSS.

6. Conclusions

This paper presents improvement of power systems low frequency oscillations using HBMO based designed PSS in a SMIB power system. The PSS design problem is converted into an optimization problem which is solved by a HBMO technique with the time domain-based objective function considering different operation conditions. The proposed HBMO algorithm combines the advantages of GA, SA, local search, and some innovations for its self-adaptation. It has stronger global search ability and more robust than GA and other heuristic methods. The effectiveness of the proposed stabilizer, for power system stability improvement, is demonstrated by a weakly connected example power system subjected to severe disturbance in comparison with GA and classical methods based designed PSS to show its superiority. The nonlinear simulation results under wide range of operating conditions show the robustness of the proposed method and their ability to provide efficient damping of low frequency oscillations. The system performance characteristics in terms of ITAE and FD indices reveal that using the proposed stabilizer the dynamic performance of the system such as overshoot, undershoot and settling time are greatly reduced under various severe disturbances.

Appendix: System data

Generator: R_a=0, x_d= 2.0, x_q = 1.91, x'_d=0.244, x'_q=0.244 , f = 50 Hz, T'_{do}=4.18, T'_{qo}=0.75, H=3.25,

Transmission line: R= 0, x_e = 0.3.

Exciter: K_A=50, T_A=0.05, E_{fdmax}=7.0, E_{fdmin}=-7.0.

References

[1] M. Anderson and A. A. Fouad, Power System Control and Stability, Ames, IA: Iowa State Univ. Press, 1977.

[2] P. Kundur, Power System Stability and Ccontrol, McGraw-Hill Inc., New York, 1994.

[3] Y. Hsu, C. Y. Hsu, Design of proportional integral power system stabilizer, IEEE Trans. on Power Systems, Vol. 1, No. 2, pp. 46-53, 1986.

[4] P. Kundur, M. Klein, G. J. Rogers, M. S. Zywno, Application of power system stabilizers for enhancement of overall system stability, IEEE Trans. on Power Systems, 614-626, 1989.

[5] M. J. Gibbard, Robust design of fixed-parameter power system stabilizers over a wide range of operating conditions, IEEE Trans. on Power Systems, Vol. 6, pp. 794-800, 1991.

[6] D. A. Pierre, A perspective on adaptive control of power systems, IEEE Trans. on Power Systems, Vol. 2, pp. 387-396, 1987.

[7] Y. Zhang, G. P. Chen, O. P. Malik and G. S. Hope, An artificial neural network based adaptive power system stabilizer, IEEE Trans. on Energy Conversion, Vol. 8, No. 1, pp. 71-77, 1993.

[8] Y. L. Abdel-Magid, M. A. Abido, Optimal multiobjective design of robust power system stabilizers using genetic algorithms, IEEE Trans.

on Power Systems, Vol. 18, No. 3, pp. 1125 - 1132, 2003.
[9] H. Shayeghi, H.A. Shayanfar, S. Jalilzadeh, A. Safari, Multi-machine power system stabilizers design using chaotic optimization algorithm, Energy Conversion and Management, Vol. 51, pp. 1572-1580, 2010.
[10] S. Mishra, M. Tripathy, J. Nanda, Multi-machine power system stabilizer design by rule based bacteria foraging, Electric Power Systems Research, Vol. 77 pp. 1595-1607, 2007.
[11] H. Shayeghi, H.A. Shayanfar, A. Safari, R. Aghmasheh, A robust PSSs design using PSO in a multi-machine environment, Energy Conversion and Management, Vol. 51, pp. 696-702, 2010.
[12] O. Bozorg Haddad, A. Afshar, M. A. Mariño, Honey-bees mating optimization (HBMO) algorithm: a new heuristic approach for water resources optimization, Water Resources Managament, Vol. 20, pp. 661-680, 2006.
[13] F. Mohammad, A. Babak, M. Ali, Application of honey-bee mating optimization algorithm on clustering, Applied Mathematics and Computations, Vol., 1502-1513, 2007.
[14] K. R. Padiyar, Power system dynamics- stability and control, Second edition, BS Publications, Hyderabad, India, 2008.
[15] B. Amiri, M. Fathian, Integration of self organizing feature maps and honey bee mating optimization algorithm for market segmentation, Journal of Theoretical and Applied Information Technology, pp. 70-86, 2007.
[16] M. Fathian, B. Amiri, A honeybee-mating approach for cluster analysis, International Journal of Advance Manufacturing and Technology, Vol. 38, pp. 809-821, 2008.
[17] H. Shayeghi, A. Jalili, H. A. Shayanfar, Multi-stage fuzzy load frequency control using PSO, Energy Conversion and Management, Vol. 49, pp. 2570-2580, 2008.

Biographies

Hossein Shayeghi received the B.S. and M.S.E. degrees in Electrical and Control Engineering in 1996 and 1998, respectively. He received his Ph. D. degree in Electrical Engineering from Iran University of Science and Technology, Tehran, Iran in 2006. Currently, he is an Associate Professor in Technical Engineering Department of University of Mohaghegh Ardabili, Ardabil, Iran. His research interests are in the Application of Robust Control, Artificial Intelligence and Heuristic Optimization Methods to Power System Control Design, Operation and Planning and Power System Restructuring. He has authored five books in Electrical Engineering area (all in Persian language). He has published more than 160 papers in International Journals and Conferences proceedings. He is a member of Iranian Association of Electrical and Electronic Engineers (IAEEE) and IEEE.

Heidarali Shayanfar Received the B.S. and M.S.E. Degrees in Electrical Engineering in 1973 and 1979, respectively. He received his Ph. D. Degree in Electrical Engineering from Michigan State University, U.S.A., in 1981. Currently, He is a Full Professor in Electrical Engineering Department of Iran University of Science and Technology, Tehran, Iran. His Research Interests are in the Area of Application of Artificial Intelligence to Power System Control Design, Dynamic Load Modeling, Power System Observability Studies, Voltage Collapse, Congestion Management in a Restructured Power System, Reliability Improvement in Distribution Systems and Reactive Pricing in Deregulated Power Systems. He has published more than 350 papers in International Journals and Conferences proceedings. He is a Member of the Iranian Association of Electrical and Electronic Engineers and IEEE. He has published more than 350 techincal papers in the International Journals and Conferences Proceedings.

Adel Akbarimajd was born in 1975 and received his B.S. degree in Control Engineering from Ferdousi University of Mashhad, Iran in 1997 and M.S. degree in Control Systems from Tabriz University of Iran in 2000. In 2007, he spent six month in the Biorobotics Laboratory, EPFL, Switzerland as visiting researcher. He received his Ph.D. in AI and robotics from University of Tehran in 2009. Later he moved to Department of Engineering at University of Mohaghegh Ardabili, where he currently is an assistant professor. His research interests include dynamical systems, control theory, system identification and computational intelligence.

Ali Ghasemi Received the B.S. Degree in Electrical Engineering from Esfahan University of Technology, Esfan, Iran in 2009. Currently, He is a M.S.E. student in Technical Eng. Department of the University of Mohaghegh Ardabili, Ardabil, Iran. His Areas of Interest in Research are the Application of Heuristic Optimization to Power System Control.

An Automated Derivation of Łukasiewicz's *CN* from Frege's Sentential Calculus

Jack K. Horner
P. O. Box 266
Los Alamos, New Mexico 87544 USA

Abstract

Two logics are implicationally equivalent if the axioms and inference rules of each imply the axioms of the other. Characterizing the inferential equivalences of various formulations of the sentential calculi is foundational. Using an automated deduction system, I show that Łukasiewicz's CN can be derived from Frege's Begriffsschrift, *the first sentential calculus ; the proof appears to be novel.*

Keywords: propositional logic, automated deduction, sentential calculus

1.0 Introduction

From Aristotle to the present it has been widely held that the equivalent of a propositional, or sentential, logic (also known as a propositional or sentential "calculus"), is required by rational reasoning. If this view of rational reasoning is correct, characterizing equivalences of various formulations of the sentential calculi is foundational to the study of logic ([1],[3]-[7],[9]-[10],[12]-[15]).

"CN", the formulation of the sentential calculus in [1], is among the most austere: its vocabulary contains only two logical connectives (C, and N) and sentence variables (*p*, *q*, *r*, ...). It has two inference rules (condensed detachment and substitution), and three axioms.

In CN, any expression of the form *Cxy* or *Nz*, where *x*, *y*, and *z* are sentences, is a sentence. *Cpq* is interpreted as "sentence *p* implies sentence *q*"; *Np* is interpreted as "not-*p*". N has higher associative precedence than C. For example,

CCqrCpNr

translates to the more common "arrow-and-parenthesis" notation as

$$(q \rightarrow r) \rightarrow (p \rightarrow \sim r)$$

where "$\rightarrow$" designates "implies" and "~" designates "not".

The axioms of CN in [1] are:

CN1. *CCpqCCqrCpr*
CN2. *CCNppp*
CN3. *CpCNpq*

Cast in CN notation, the axioms of Frege's *Begriffsschrift* (BG, [10]), the first sentential calculus, are

CN18. *CqCpq*
CN21. *CCpCqrCqCpr*
CN35. *CCpCqrCCpqCpr*
CN39. *CNNpp*
CN40. *CpNNp*
CN46. *CCpqCNqNp*

The main result of this paper is that [10] implies [1].

2.0 Method

To show that [10] implies [1], the *prover9* ([2]) script shown in Figures 1 was executed under on a Dell Inspiron 545 with an Intel Core2 Quad CPU Q8200 @ 2.33 GHz and 8.00 GB RAM, running under the *Windows Vista Home Premium (SP2)/Cygwin* operating environment.

```
set(hyper_resolution).

formulas(usable).
P ( i(y, i(x,y)) )                          # label("CN18").
P ( i(i(x, i(y,z)), i(y, i(x,z))) )         # label("CN21").
P ( i(i(x, i(y,z)), i(i(x,y), i(x,z))) )    # label("CN35").
P ( i((-(-x)), x) )                         # label("CN39").
P ( i(x, -(-x)) )                           # label("CN40").
P ( i(i(x,y), i(-y, -x)) )                  # label("CN46").
end_of_list.

formulas(sos).
-P(i(x,y)) | -P(x) | P(y)           # label("InfConDet").
end_of_list.

formulas(goals).
P( i(i(x,y), i(i(y,z), i(x,z))) )       # label("AxCN1").
P( i(i(-x,x), x) )                      # label("AxCN2").
P( i(x, i(-x,y)) )                      # label("AxCN3").
end_of_list.
```

Figure 1. The *prover9* script used to show that BG implies CN. The implementation of condensed detachment is the formula in the "sos" list; substitution is derived from *prover9*'s hyperresolution rule (introduced in the "set" command at the top of the script). Details of *prover9*'s syntax and semantics can be found in [2].

3.0 Results

Figure 2 shows BG implies CN.

```
============================== PROOF =================================

% Proof 1 at 1.06 (+ 0.12) seconds: "AxCN2".

2 P(i(i(-x,x),x)) # label("AxCN2") # label(non_clause) # label(goal).
[goal].
4 P(i(x,i(y,x))) # label("CN18").  [assumption].
5 P(i(i(x,i(y,z)),i(y,i(x,z)))) # label("CN21").  [assumption].
```

```
6 P(i(i(x,i(y,z)),i(i(x,y),i(x,z)))) # label("CN35").  [assumption].
7 P(i(--x,x)) # label("CN39").  [assumption].
8 P(i(x,--x)) # label("CN40").  [assumption].
9 P(i(i(x,y),i(-y,-x))) # label("CN46").  [assumption].
10 -P(i(x,y)) | -P(x) | P(y) # label("InfConDet").  [assumption].
12 -P(i(i(-c4,c4),c4)) # label("AxCN2") # answer("AxCN2").  [deny(2)].
26 P(i(i(i(x,i(y,z)),i(x,y)),i(i(x,i(y,z)),i(x,z)))).
[hyper(10,a,6,a,b,6,a)].
29 P(i(-x,i(i(y,x),-y))).  [hyper(10,a,5,a,b,9,a)].
33 P(i(x,i(y,y))).  [hyper(10,a,5,a,b,4,a)].
35 P(i(x,i(y,--y))).  [hyper(10,a,4,a,b,8,a)].
36 P(i(x,i(--y,y))).  [hyper(10,a,4,a,b,7,a)].
43 P(i(x,x)).  [hyper(10,a,33,a,b,33,a)].
49 P(i(x,i(i(x,y),y))).  [hyper(10,a,5,a,b,43,a)].
74 P(i(i(x,--y),i(x,y))).  [hyper(10,a,6,a,b,36,a)].
91 P(i(i(i(x,i(i(x,y),y)),z),z)).  [hyper(10,a,49,a,b,49,a)].
204 P(i(i(-x,i(y,x)),i(-x,-y))).  [hyper(10,a,6,a,b,29,a)].
205 P(i(x,i(-y,i(i(z,y),-z)))).  [hyper(10,a,4,a,b,29,a)].
353 P(i(i(x,i(--x,y)),i(x,y))).  [hyper(10,a,26,a,b,35,a)].
10023 P(i(x,i(i(y,-x),-y))).  [hyper(10,a,353,a,b,205,a)].
10451 P(i(i(x,-y),i(y,-x))).  [hyper(10,a,5,a,b,10023,a)].
12892 P(i(-x,-i(-x,x))).  [hyper(10,a,91,a,b,204,a)].
12916 P(i(i(-x,x),--x)).  [hyper(10,a,10451,a,b,12892,a)].
13014 P(i(i(-x,x),x)).  [hyper(10,a,74,a,b,12916,a)].
13015 $F # answer("AxCN2").  [resolve(13014,a,12,a)].

============================== end of proof ==========================


============================== PROOF =================================

% Proof 2 at 2.96 (+ 0.33) seconds: "AxCN1".

1 P(i(i(x,y),i(i(y,z),i(x,z)))) # label("AxCN1") # label(non_clause) #
label(goal).  [goal].
4 P(i(x,i(y,x))) # label("CN18").  [assumption].
5 P(i(i(x,i(y,z)),i(y,i(x,z)))) # label("CN21").  [assumption].
6 P(i(i(x,i(y,z)),i(i(x,y),i(x,z)))) # label("CN35").  [assumption].
10 -P(i(x,y)) | -P(x) | P(y) # label("InfConDet").  [assumption].
11 -P(i(i(c1,c2),i(i(c2,c3),i(c1,c3)))) # label("AxCN1") #
answer("AxCN1").  [deny(1)].
26 P(i(i(i(x,i(y,z)),i(x,y)),i(i(x,i(y,z)),i(x,z)))).
[hyper(10,a,6,a,b,6,a)].
37 P(i(x,i(i(y,i(z,u)),i(i(y,z),i(y,u))))).  [hyper(10,a,4,a,b,6,a)].
39 P(i(x,i(y,i(z,y)))).  [hyper(10,a,4,a,b,4,a)].
351 P(i(i(x,i(i(y,x),z)),i(x,z))).  [hyper(10,a,26,a,b,39,a)].
7590 P(i(i(x,y),i(i(z,x),i(z,y)))).  [hyper(10,a,351,a,b,37,a)].
21088 P(i(i(x,y),i(i(y,z),i(x,z)))).  [hyper(10,a,5,a,b,7590,a)].
21089 $F # answer("AxCN1").  [resolve(21088,a,11,a)].

============================== end of proof ==========================

============================== PROOF =================================

% Proof 3 at 3.62 (+ 0.42) seconds: "AxCN3".
```

```
3 P(i(x,i(-x,y))) # label("AxCN3") # label(non_clause) # label(goal).
[goal].
4 P(i(x,i(y,x))) # label("CN18").  [assumption].
5 P(i(i(x,i(y,z)),i(y,i(x,z)))) # label("CN21").  [assumption].
6 P(i(i(x,i(y,z)),i(i(x,y),i(x,z)))) # label("CN35").  [assumption].
7 P(i(--x,x)) # label("CN39").  [assumption].
8 P(i(x,--x)) # label("CN40").  [assumption].
9 P(i(i(x,y),i(-y,-x))) # label("CN46").  [assumption].
10 -P(i(x,y)) | -P(x) | P(y) # label("InfConDet").  [assumption].
13 -P(i(c5,i(-c5,c6))) # label("AxCN3") # answer("AxCN3").  [deny(3)].
26 P(i(i(i(x,i(y,z)),i(x,y)),i(i(x,i(y,z)),i(x,z)))).
[hyper(10,a,6,a,b,6,a)].
29 P(i(-x,i(i(y,x),-y))).  [hyper(10,a,5,a,b,9,a)].
34 P(i(x,i(i(y,z),i(-z,-y)))).  [hyper(10,a,4,a,b,9,a)].
35 P(i(x,i(y,--y))).  [hyper(10,a,4,a,b,8,a)].
36 P(i(x,i(--y,y))).  [hyper(10,a,4,a,b,7,a)].
39 P(i(x,i(y,i(z,y)))).  [hyper(10,a,4,a,b,4,a)].
74 P(i(i(x,--y),i(x,y))).  [hyper(10,a,6,a,b,36,a)].
205 P(i(x,i(-y,i(i(z,y),-z)))).  [hyper(10,a,4,a,b,29,a)].
351 P(i(i(x,i(i(y,x),z)),i(x,z))).  [hyper(10,a,26,a,b,39,a)].
353 P(i(i(x,i(--x,y)),i(x,y))).  [hyper(10,a,26,a,b,35,a)].
532 P(i(-i(x,y),-i(x,--y))).  [hyper(10,a,9,a,b,74,a)].
7591 P(i(x,i(-x,-y))).  [hyper(10,a,351,a,b,34,a)].
7621 P(i(-x,i(x,-y))).  [hyper(10,a,5,a,b,7591,a)].
7861 P(i(-i(x,-y),--x)).  [hyper(10,a,9,a,b,7621,a)].
8920 P(i(-i(x,-y),x)).  [hyper(10,a,74,a,b,7861,a)].
8963 P(i(x,i(-i(y,-z),y))).  [hyper(10,a,4,a,b,8920,a)].
9784 P(i(i(x,-i(y,-z)),i(x,y))).  [hyper(10,a,6,a,b,8963,a)].
10023 P(i(x,i(i(y,-x),-y))).  [hyper(10,a,353,a,b,205,a)].
10451 P(i(i(x,-y),i(y,-x))).  [hyper(10,a,5,a,b,10023,a)].
21813 P(i(-i(x,y),x)).  [hyper(10,a,9784,a,b,532,a)].
21836 P(i(x,--i(-x,y))).  [hyper(10,a,10451,a,b,21813,a)].
22070 P(i(x,i(-x,y))).  [hyper(10,a,74,a,b,21836,a)].
22071 $F # answer("AxCN3").  [resolve(22070,a,13,a)].

============================== end of proof ==========================
```

Figure 2. Summary of a *prover9* ([2]) proof showing that BG ([10]) implies CN ([1]).

The total time to complete the proofs shown in Figure 2 was ~9 seconds on the platform described in Section 2.0.

4.0 Conclusions and discussion

Section 3 demonstrates that BG implies CN. A companion paper ([16]) proves CN implies BG.

The proof in Figure 2 appears to be novel.

5.0 References

[1] Łukasiewicz J. *Elements of Mathematical Logic*. Second Edition (1958). Trans. by Wojtasiewicz O. Pergamon Press. 1963.

[2] McCune WW. *prover9 and mace4.* http://www.cs.unm.edu/~mccune/prover9/. 2009.

[3] Aristotle. *Prior Analytics*. Trans. by A. J. Jenkinson. In Aristotle. *The Basic Works of Aristotle.* Ed. by R. McKeon. Random House. 1941. pp. 62-107.

[4] Aristotle. *Posterior Analytics.* Trans. by G. R. G. Mure. In Aristotle. *The Basic Works of Aristotle.* Ed. by R. McKeon. Random House. 1941. pp. 108-186.

[5] Tarski A. *Introduction to Logic*. Trans. by O. Helmer. Dover. 1941.

[6] Hempel C. Studies in the logic of explanation. In Hempel C. *Aspects of Scientific Explanation and Other Essays in the Philosophy of Science.* Free Press. 1965. pp. 245-290.

[7] Quine WVO. *Philosophy of Logic*. Second Edition. Harvard. 1986.

[8] Chang CC and Keisler HJ. *Model Theory*. North-Holland. 1990.

[9] Russell B and Whitehead AN. *Principia Mathematica*. Volume I (1910). Merchant Books. 2009.

[10] Frege G. Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens. Halle. 1879. Translated in van Heijenoort J. Begriffsschrift, a formula language, modeled upon that of arithmetic, for pure thought. *From Frege to Gödel: A Source Book in Mathematical Logic, 1879-1931*. Harvard. 1967. pp. 3-82.

[11] Horn A. On sentences which are true of direct unions of algebras. *Journal of Symbolic Logic* 16 (1951), 14–21.

[12] Church A. *Introduction to Mathematical Logic.* Volume I. Princeton. 1956.

[13] Birkhoff G and von Neumann J. The logic of quantum mechanics. *Annals of Mathematics* 37 (1936), 823-843.

[14] Kant I. *Kant's Introduction to Logic* (1800). Trans. by Abbott TK. Greenwood Press. 1963.

[15] Cohen MR and Nagel E. *An Introduction to Logic and Scientific Method.* Harcourt, Brace, and Company. 1934.

[16] Horner JK. An automated derivation of Frege's sentential calculus from Łukasiewicz's. *Proceedings of the 2011 International Conference on Artificial Intelligence.* CSREA Press. Forthcoming.

TRANSFORMATION TO NEAR GAUSSIAN DISTRIBUTION IN FEATURE SPACE BASED ON KERNEL PCA

Ying-Can Wei
School of Mathematics and Applied Physics , Nanjing University of Posts and Telecommunications, Nanjing , Jiangsu , China

Abstract *- Kernel Principal Components Analysis (KPCA) algorithm is more popular for feature extraction and data classification in recent years. However, KPCA is limited only to the Gaussian distribution of the data feature extraction better. To solve this problem,based on the theory of Applied Multivanriate Statistical Analysis, we propose a new and efficient algorithm called TNG (Transformation to Near Gaussian) power transformation for data so as to find the optimal kernel function. Amied at making mapping data in feature space near Gaussian distribution. Through One-dimensional and Muliti-dimensional data exprimental results show that TNG algorithm efficiently improve Kernel PCA feature extraction and SVM's performance on the accuracy of classification problems.*

Keywords: Multivanriate Statistical Analysis; Feature subspace; Power transformation; Gaussian distribution; Optimal kernel function

1 Introduction

Kernel Principal Components Analysis (Kemel PCA) [1]is a nonlinear PCA extansion, Suppose we first map the data X that nonlinear and obey any distribution in input space into a feature space F by kernel function φ, we can still perform PCA in feature space F [2], then extract eigenvalues of the data.

Support Vector Machines(SVM) is a learning algorithm created by the Vapnik [3] based on statistical learning theory, it has outstanding advantage of generalization ability and so on. Fortunately, today KPCA is successful application with SVM in classification and regression problems and makes greatly improved on accuracy.

Since PCA [2] feature extraction on the premise that only the Gaussian distribution of the data has a better feature extraction effect. So in the feature space mapping data near obey Gaussian distribution estimate, for the performance of KPCA feature extraction and SVM classification, accuracy of regression problems has a vital role.

In this paper, according to data space Gaussian distribution estimate, we propose the TNG power Transformation algorithm based on the theory of Applied Multivanriate Statistical Analysis [6]. Our method has three main step: First, high-dimension feature space F for the Non-Gaussian distribution of data using appropriate TNG power transformation, Then through the optimal power transformation parameter optimize Kernel function, so that the Non-Gaussian mapping data more "towards Gaussian". Thirdly, by adopting TNG algorithm, the input space data mapped by optimalized Kernel function, we can implement Gaussian distribution theoretic analysis for mapping data in feature spce.

Finally, exprimental results proved Knernel PCA-SVM classification algorithm optimalized by TNG power transformation method has an better performance than other same type algorithms.

2 Backdround

2.1 KPCA

Assuming observed object has M kind of features. In expriment, given a set of observed sample x_k , $k = 1,..., M, x_k \in R^N$. First, our sample mapped into feature space F by a nonlinear function φ , $\varphi : R^N \rightarrow F$, $x \rightarrow \varphi(x)$. The mapping sample in F is centered [7][8], i.e. $\sum_{k=1}^{M} \boldsymbol{\varphi}(x_k) = 0$. To do PCA for the covariance matrix :

$$\mathbf{C} = \frac{1}{M} \sum_{j=1}^{M} \boldsymbol{\varphi}(x_j)\, \boldsymbol{\varphi}(x_j)^T$$

We have to find the Eigenvalues $\lambda \geq 0$ of C and Eigenvectors $V \in F$ satisfying $\mathbf{CV} = \lambda \mathbf{V}$. Substituting aboved formula, we note that all solutions lie in the span of $\mathbf{V} \in \mathbf{span}\{\boldsymbol{\varphi}(x_1),..., \boldsymbol{\varphi}(x_M)\}$,and there exist coefficient $\alpha_i (i = 1,..., M)$ such that :

$$\mathbf{V} = \sum_{i=1}^{M} \alpha_i \boldsymbol{\varphi}(x_i) \qquad (1)$$

We arrive at [1]:

$$\lambda\sum_{i=1}^{M}\alpha_i(\varphi(x_k)\cdot\varphi(x_i))=\frac{1}{M}\sum_{i=1}^{M}\alpha_i(\varphi(x_k)\cdot\sum_{j=1}^{M}\varphi(x_j))(\varphi(x_j)\cdot\varphi(x_i)) \quad (2)$$

and $k=1,...,M$

Through defining an $M\times M$ Kernel matrix by $\mathbf{K}_{i,j}=(\boldsymbol{\varphi}(x_i)\cdot\boldsymbol{\varphi}(x_j))=\mathbf{k}(x_i,x_j)$,formula (2) translates into [7]:

$$\mathbf{K}\alpha=M\lambda\alpha \quad (3)$$

Therefor, to determine Eigenvectors $\mathbf{V}$ and calculating coefficients $\alpha_i(i=1,...,M)$ problem only rely on the kernel matric $\mathbf{K}$ Eigenvalues decomposition. For any test sample $\mathbf{x}$, it maps the feature space F point $\boldsymbol{\varphi}(\mathbf{x})$. According to

$$(\mathbf{V}^k\cdot\boldsymbol{\varphi}(\mathbf{x}))=\sum_{i=1}^{M}\alpha_i^k(\boldsymbol{\varphi}(x_i)\cdot\boldsymbol{\varphi}(\mathbf{x})) \quad (4)$$

We compute projections of a test point $\boldsymbol{\varphi}(\mathbf{x})$ onto the Eigenvectors $\mathbf{V}$ in F and extract the nonlinear principal component of mapping data $\boldsymbol{\varphi}(\mathbf{x})$.

2.2 SVM

Consider a pattern classification, which uses a hyperplane to separate two classes based on given data: $\{(x_i, y_i),\ x_i\in R^n,\ y_i\in(-1,1)),\ i=1,2\cdots n\}$. First, a nonlinear SVM maps the input data X into a high dimensional feature space by using a nonlinear mapping φ.Then construct a optimal separating hyperplane [5][9]:

$$y(x)=\mathrm{sgn}(\omega\bullet\varphi(x)+b),$$

And ω is weigh vector, b is the bias(or –b is the threshold). According to functional theory, When a Kernel function $k(x_i,x_j)$satisfy Mercer conditions, it corresponding a transformation space of inner product, that is:

$$k(x_i,x_j)=(\varphi(x_i),\varphi(x_j)) \quad (5)$$

Thus SVM algorithm is transformed into solving the following original problem:

$$MaxQ(\alpha)=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K(x_i,x_j)$$

According to KKT conditions, Lagrange multiplier α_i must satisfy :

$$\alpha_i(y_i(\omega^T\varphi(x_i)+b-1+\xi_i)=0 \quad \text{i=1,2,……n}$$

α_i values are not zero corresponding the sample called support vector machine. To get the final classifier is [9]:

$$f(x)=\mathrm{sgn}\sum_{sv}\alpha_i y_i K(x_i,x_j)+b \quad (6)$$

Choose a diffident Kernel function $K(x_i,x_j)$, we can construct diffident SVM.

3 Previous works

3.1 Structure feature subspace

First, Kernel PCA feature space $\mathbf{F}$ coordinate systems to do orthogonal transform and get the new axes $\{\mathbf{V}_1,\mathbf{V}_2,...,\mathbf{V}_M\}$, M is observation number.

If formula (3) non-zero Eigenvalues in descending order $\lambda_1\geq\lambda_2\geq...\geq\lambda_p\geq0$, corresponding to p Eigenvectors $\{\mathbf{V}_1,\mathbf{V}_2,...,\mathbf{V}_p\}$ constitute a subspace $\mathbf{F}^p$ of feature space $\mathbf{F}$, called feature scuspace. According to know [1]: All the $\{\mathbf{V}_1,\mathbf{V}_2,...,\mathbf{V}_p\}$ lie in the span of $\{\boldsymbol{\varphi}(x)\}$ and they form a subset of the base vector sets in feature space. Therefor any $\boldsymbol{\varphi}(\mathbf{x})$ can use $\{\mathbf{V}_1,\mathbf{V}_2,...,\mathbf{V}_p\}$ as base of a set with coordinates $\{\beta_1,\beta_2,...\beta_p\}$approximate denotation $\hat{\boldsymbol{\varphi}}(\mathbf{x})$, thus expression can be expressed [1][7][8][10]:

$$\boldsymbol{\varphi}(x)\cong\hat{\boldsymbol{\varphi}}(x)=\sum_{k=1}^{p}\beta_k\mathbf{V}_k \quad (7)$$

Since the data set $\{\boldsymbol{\varphi}(x)\}$ can be expressed as $\{\hat{\boldsymbol{\varphi}}(x)\}$, and $\{\mathbf{V}_1,\mathbf{V}_2,...,\mathbf{V}_p\}$ is a set of orthogonal basis, and only the base coordinate systems do orthogonal transform and discard the redundant set. In addition, from the Gaussian distribution of the linear additivity principle that we can get data distribution in the course of series keep the same characteristics of distribution. If $\{\hat{\boldsymbol{\varphi}}(x)\}$ obey Gaussian distribution, the $\boldsymbol{\beta}(\mathbf{x})$ that is projections of $\hat{\boldsymbol{\varphi}}(\mathbf{x})$ in feature Subspace $\mathbf{F}^p$, has the same distribution with $\hat{\boldsymbol{\varphi}}(\mathbf{x})$. Based on this, in order to estimate the distribution of $\boldsymbol{\varphi}(\mathbf{x})$ in feature space, we equivalent to analysis the distribution of $\boldsymbol{\beta}(\mathbf{x})$ in feature Subspace $\mathbf{F}^p$. We pay special attention to $\boldsymbol{\beta}(\mathbf{x})$ is a vector composed of the principal component of $\mathbf{x}$ in each feature directions projections coordinate. It is only one we can calculate the value. This also prove that our method is feasible.

3.2 The Theoretic Basis of TNG Algorithm

According to Applied Multivariate Statistical Analysis theory [6], we can draw that transformations are nothing more than than a reexpression of the data in different units. For example, when a histogram of positive observation exhibits a long right-hand tail, transforming the observation by taking their logarithms or square roots will often markely improve the symmety about the mean and the approximation to a Gaussian distribution. It has been shown theoretically that data that counts can often be made of more Gaussian by taking their *square roots.* Similarly, the *logit transformation* applied to proportions and *Fisher's Z*-transformation applied to correlation coefficients yield quantities that are approximately Gaussian distributed [7].

For this reason, in many instances, the choice of a transformation to improve the approximation to Gaussian is not obvious. For such cases, it is convenient to let the data suggest a transformation. A useful family of transformation for this purpose is the family of *power transformation*.

Let X represent an arbitrary observation. The power family of transformations is indexed by a parameter λ. A given value for λ implies a particular transformation. We can trace the family of transformations as λ ranges from negative to positive power of x. A sequence of possible transformations is:

$$\underbrace{\cdots x^{-1}=\frac{1}{x}, x^{0}=\ln x, x^{\frac{1}{4}}=\sqrt[4]{x},}_{\text{shrinks large values of x}} \qquad \underbrace{x^{2}, x^{3}, \cdots}_{\text{Increases large values of x}}$$

To select a power transformation, an investigator final choice should always be examined by a Q-Q plot. This also is Multivariate Statistical Analysis theory in expriments often use inspection method. Q-Q plot describe the Gaussian quantile $q_{(j)}$ versus the ordered data $x_{(j)}$ one would expect to observe if the observation actually were Gaussian distributed. It can be used to assess the assumption of Gassian. When the point （ $q_{(j)}$, $x_{(j)}$ ） very nearly along a straight line, the Gaussian assumption remains tenable.

Based on the above mentioned, A convenient analytical method is available for choosing a power transformation. We begin by focusing our attention on the *univariate* case. Box and Cox [11] consider the slightly modified family of power transformations:

$$x^{\lambda}=\begin{cases}\dfrac{x^{\lambda}-1}{\lambda} & \lambda \neq 0\\ \ln x & \lambda = 0\end{cases} \qquad (8)$$

Which is continuous in λ for $x \succ 0$ [12], Given the observations $x_1, x_2, \cdots x_n$, the Box-Cox solution for the choice of an appropriate power λ is the solution that *maximazes* the expression:

$$\ell(\lambda)=-\frac{n}{2}\ln[\frac{1}{n}\sum_{j=1}^{n}(x_j^{(\lambda)}-\overline{x^{(\lambda)}})^2]+(\lambda-1)\sum_{j=1}^{n}\ln x_j \qquad (9)$$

We note that $x_j^{(\lambda)}$ is defined in（8）and

$$\overline{x^{(\lambda)}}=\frac{1}{n}\sum_{j=1}^{n}x_j^{(\lambda)}=\frac{1}{n}\sum(\frac{x_j^{\lambda}}{\lambda}) \qquad (10)$$

is the arithmetic average of the transformed observations.

The first term in formula (9) is, apart from a constant, the logarithm of a Gaussian likelihood function, after maximizing it with respect to to the population mean and variance parameters. The calculation of $\ell(\lambda)$ for many values of λ is an easy task for a computer. It helpful to have a graph of both to find the final solution.

A EXAMPLE:

We gave readings of the microwave radiation emitted through the *closed doors* of n=100. Since each data contains 15 features, we only select one feature value as research object. The Q-Q plot of these data in Figure 1 indicates that the observations deviate from what would be expected if they were Gaussian distributed. Let us perform a power transformation of the data which, we hope, will produce results that are more nearly Gassian. Restricting our attention to the family of transformation in formula (8), we must find that value of λ maximizing the function $\ell(\lambda)$ in formula (9).

So we made λ and $\ell(\lambda)$ curve graphics, it allows the more the exact determination $\hat{\lambda}=0.28$ is show in Figure 2.

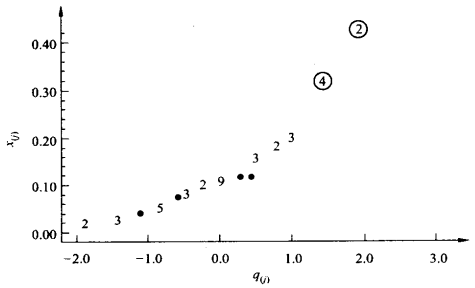


Figure 1.A Q-Q plot of the radiation data(door closed)

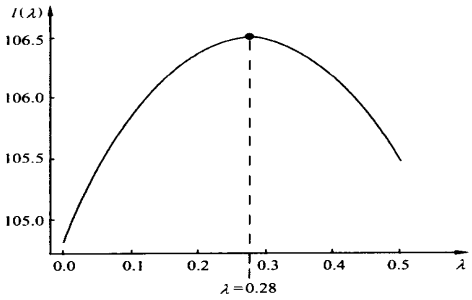


Figure 2.Plot of $\ell(\lambda)$ versus λ for radiation data.

It is evident from the plot that a value of $\hat{\lambda}$ around 0.30 maximazes $\ell(\lambda)$. For convenience, we choose $\hat{\lambda}=0.25$. The data x_j were reexpressed as:

$$x_j^{1/4}=\frac{x_j^{1/4}-1}{1/4} \qquad \text{j=1,2,}\cdots\text{, 100}$$

and a Q-Q plot was constructed from the transformed quantities. This plot is shown in Figure 3. The quantile pairs fall very close to a straight line, and we would conclude from this evidence that the $x_j^{1/4}$ are near Gassian.

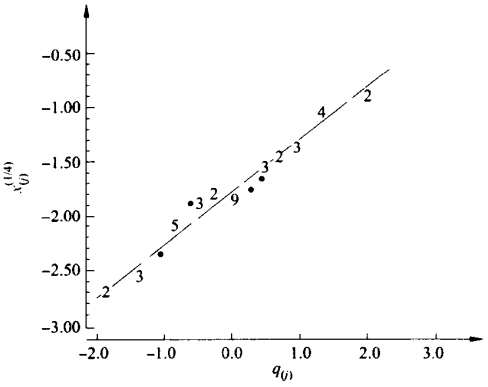


Figure 3.A Q-Q plot of transformation radiation data.

Muliti-dimensional Data Power Transformation

But in reality situation,data always Muliti-dimensional. Therefor [12], the implementation of Muliti-dimensional observations power transformation to approximate Gaussian distribution is of great significance. Now we are derived for the Muliti-dimensional data power transformation based on univariate case. With multivariate observations, a power transformation must be selected for each of the variables. Let $\lambda_1, \lambda_2, \cdots \lambda_p$ be power transformations for the p measured characteristics. Each λ_k can be selected by maximizing [4][11]:

$$\ell_k(\lambda) = -\frac{n}{2}\ln[\frac{1}{n}\sum_{j=1}^{n}(x_{jk}^{(\lambda_k)} - \overline{x_k^{\lambda_k}})^2] + (\lambda_k - 1)\sum_{j=1}^{n}\ln x_{jk} \qquad (11)$$

Where $x_{1k}, x_{2k}, \cdots, x_{nk}$ are the n observations on the kth variable, $k = 1,2,\cdots p$. Here

$$\overline{x_k^{(\lambda_k)}} = \frac{1}{n}\sum_{j=1}^{n} x_{jk}^{(\lambda_k)} = \frac{1}{n}\sum(\frac{x_{jk}^{\lambda_k}}{\lambda_k}) \qquad (12)$$

is the arithmetic of the transformation observations. The jth transformed multivariate observation is:

$$x_j^{(\hat{\lambda})} = [\frac{x_{j1}^{\hat{\lambda}_1} - 1}{\hat{\lambda}_1}, \frac{x_{j2}^{\hat{\lambda}_2} - 1}{\hat{\lambda}_2} \cdots, \frac{x_{jp}^{\hat{\lambda}_p} - 1}{\hat{\lambda}_p}]^T \qquad (13)$$

Where $\hat{\lambda}_1, \hat{\lambda}_2, \cdots \hat{\lambda}_p$ are the value that individually maximize formula (11).

The procedure just described is equivalent to making each marginal distribution approximately Gassian. Although Gaussian marginals are not sufficient to ensure that the joint distribution is Gassian, in practical applications this may be good enough. If not, we could start with the values $\hat{\lambda}_1, \hat{\lambda}_2, \cdots \hat{\lambda}_p$ obtained from the preceding transformations and iterate the set of values $\lambda' = [\lambda_1, \lambda_2, \cdots, \lambda_p]$, which collectively maximizes:

$$\ell(\lambda_1, \lambda_2, \cdots \lambda_p) = -\frac{n}{2}\ln|S(\lambda)| + (\lambda_1 - 1)\sum_{j=1}^{p}\ln x_{j1} + \cdots$$

$$+ (\lambda_2 - 1)\sum_{j=1}^{p}\ln x_{j2} + \cdots + (\lambda_p - 1)\sum_{j=1}^{p}\ln x_{jp} \qquad (14)$$

Where $S(\lambda)$ is the sample covariance matrix computed from:

$$x_j^{(\lambda)} = [\frac{x_{j1}^{\lambda_1} - 1}{\lambda_1}, \frac{x_{j2}^{\lambda_2} - 1}{\lambda_2} \cdots, \frac{x_{jp}^{\lambda_p} - 1}{\lambda_p}]^T$$

The selection method based on formula (14) is equivalent to maximizing a multivanriatelikelihood over $\mu, \sum$ and λ, whereas the method based on formula (11)corresponds to maximizing the kth univariate likelihood over $\mu_k, \sigma_{kk}, \lambda_k$ [12].

We can see that making each marginal distribution approximately Gaussian is roughly equivalent to addressing the bivariate distribution directly and making it approximately Gassian. It is generally easier to select appropriate transformations for the marginal distributions than for the joint distributions.

3.3 TNG Algorithm Proposed

Based on the aboved theoretic analysis we can formally present TNG power transformation algorithm for Non-Gaussian distributions mapping data in feature space.

Step1: Given the standard kernel parameters, extracting the principal components **β(x)** of the input space data X by formula (4) and constructed the feature subspace. Based on the theory of formula (7) proved the **β(x)** and data **φ(x)** in feature subspace distribution identity.

Step2:Since the input space data X are muliti-dimensional in reality, so modify formula (11) expressed as:

$$\ell_k(\lambda) = -\frac{p}{2}\ln[\frac{1}{p}\sum_{j=1}^{p}(\beta(x)_{jk}^{(\lambda_k)} - \overline{\beta(x)_k^{\lambda_k}})^2] + (\lambda_k - 1)\sum_{j=1}^{p}\ln\beta(x)_{jk} \qquad (15)$$

Where $\overline{\beta(x)^{(\lambda)}} = \frac{1}{p}\sum_{j=1}^{p}\beta(x)_j^{(\lambda)} = \frac{1}{p}\sum_{j=1}^{p}(\frac{\beta(x)_j^{\lambda}}{\lambda})$

$\beta(x)_{jk}$ represent projections that is the input space jth feature of kth data X in the corresponding feature Subspace vector V_k .Thus getting the optimal parameters $\hat{\lambda}_1, \hat{\lambda}_2, \cdots \hat{\lambda}_p$

Setp3： According to formula (12),feature space data expressed as:

$$\beta(x)_j^{(\hat{\lambda})} = [\frac{\beta(x)_{j1}^{\hat{\lambda}_1} - 1}{\hat{\lambda}_1}, \frac{\beta(x)_{j2}^{\hat{\lambda}_2} - 1}{\hat{\lambda}_2} \cdots, \frac{\beta(x)_{jp}^{\hat{\lambda}_p} - 1}{\hat{\lambda}_p}]^T \qquad (16)$$

Then optimal the initial kernel function $\hat{K}(x_i, x_j)$ based on formula (7) and (5), feature extraction again on the basis of **β(x)** and **φ(x)** approximately Gaussian distribution. Thus improved the ability of KPCA feature extraction.

Step4：Using the optimized kernel function $\hat{K}(x_i, x_j)$ constructed the SVM by formula (6), processing the classification and regression problems..

4 Expriments

Based on example in 3.2, radiation measurements were also recorded through the *open doors* of n=100 microwave ovens introduced. A total of 200 data ,including previous 100 data of the microwave radiation emitted through the *closed door*. Each data X contain 15 feature items, We select each data set 60% data as training set, and remaining 40% data as tese set for KPCA-SVN (TNG) algorithm. At the same as time, we also compared with KPCA-SVM、C-SVM and PCA-SVM algorithms without using TNG algorithm to optimized. In expriment, the training and testing of SVM are accomplished by LIBSVM. The initial kernel function selected for Gaussian function, kernel parameters c=0.50.

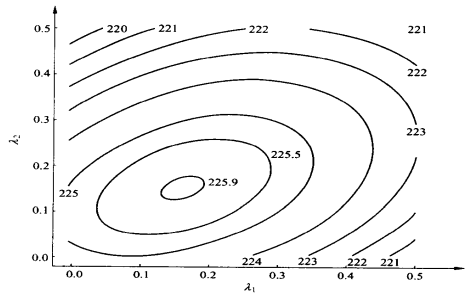


Figure 4.Contour plot of $\ell(\lambda_1, \lambda_2)$ for radiation data

Let us denote the door-closed data by $x_{1,1}, x_{2,1}, \cdots x_{100,1}$ and the door-open data by $x_{1,2}, x_{2,2}, \cdots x_{100,2}$. As each data X contain 15 feature items. Here, we consider the *joint* distribution of multivariate data and simultaneously determin the set of powers $(\lambda_1, \lambda_2, \cdots, \lambda_p)$ that makes this joint distribution approximately multivariate Gaussian distribution. p is the dimension of the feature Subspace. To do this, we must maximize $\ell(\lambda_1, \lambda_2, \cdots \lambda_p)$ in formula (14) with respect to set of powers $(\lambda_1, \lambda_2, \cdots, \lambda_p)$.

Due to the large sets data dimensions, any two dimension data selected from the data set to find the pair of powers (λ_1, λ_2). We computed $\ell(\lambda_1, \lambda_2)$ for a grid of λ_1, λ_2 values covering $0 \prec \lambda_1 \prec 0.5$ and $0 \prec \lambda_2 \prec 0.5$, and we constructed the contour plot shown in Figure 4. We see that the maximum occurs at about a pair of powers $(\hat{\lambda}_1, \hat{\lambda}_2)$ =（0.16， 0.16）. So the search for the optimal powers set $(\hat{\lambda}_1, \hat{\lambda}_2, \cdots, \hat{\lambda}_p)$ can be through the similar method.

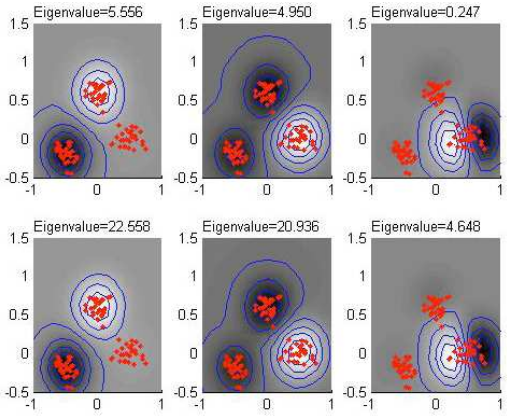


Figure 5. Two-dimensional surface contour

Adopt the TNG algorithm to find the optimal powers set, thus obtain the optimal kernel function. We selected any 30 samples from the feature space mapping data of each two data sets and projected to the two-dimension contour palne, meanwhile constructing a data set that has the same number of sample and strict obey Gaussian distribution. Compare these three dates in contour plane by clustering method in Figure 5. Through the comparative analysis we can know that the nonlinear data feature extraction is effective, and they are approximately Gaussian distribution in feature space.

Under same training set for training C-SVM，PCA-SVM and KPCA-SVM without using TNG algorithm. Through the test set to contrast-detect the several algorithm on classification performance. The results summarised in Table 1.

Table 1. Experiment Result

Algorithm	**Comparison of algorithms**		
	Feature number	***False rate (%)***	***Accuracy (%)***
C-SVM	13	5.73	94.5
PCA-SVM	11	3.15	95.55
KPCA-SVM	6	1.96	96.25
KPCA-SVM(TNG)	4	0.72	98.2

Finally carry on the comparative expriment again, the *heart data* set as the training and test data set under LIBSVM environment, and make the feature extraction quantity change in the circumstances, the error rate of several algorithm contrast graph in Figure 6. We still can see the KPCA-SVM （TNG）algorithm has obvious advantage on multi-feature extraction.

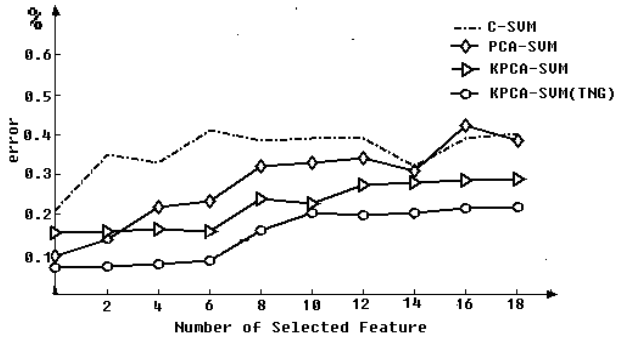


Figure 6.Plot feature number versus error rate for heart data

5 Conclusions

In this paper, We combined with the theory of Applied Multivariate Statistical Analysis to proposed a new data transformation algorithm(TNG). The purpose is to make Non-Gaussian distribution input data by choosing appropriate power transformation to get the optimal kernel function, and make mapping data in feature space approximately obey Gaussian distribution. The expriment show our method efficiently improve Kernel PCA feature extraction and SVM performance on the accuracy of classification problems. Several further work, such as theoretical analysis on the copmutational efficiency of the multiple joint distribution remain to be studied.

Acknowledgements

This work was mainly supported by the National Natural Science Foundation of China NO.61070234 and NO.61071167, which are conducted by professor LEI LI. This paper partly supported by Science and Technology Innovation Training Program NO.2010131.

Authors acknowledge professor LEI LI, who is the Dean of Mathematics and Physics, Nanjing University of Posts and Telecommunications, for he gave me a variety of comments which helped me a lot to improve the quality of my present work.

6 References

[1] Scholkopf, and Smola,"Kernel Principal Component Analysisi",Max-Planck-Institutf.biol.Tubingen,Germany GMD FIRST 1998

[2] Jonathon Shlens,"A Tutorial on Principal Component Analysis",SystemsNeurobiology Laboratory,University of California,2005

[3] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995 [4]Nello Cristianini,"An Itrodution to Support Vectory Machines and Other Kernel-based learning Methods",ISBN7-5053-9336-7, pp.24-40,98-106, Mar. 2004

[4] Andreson,D.F., and J.L. Warner. "Transformations of Multivariate Data." Biometrics, 27,no…until… (1971),825-840.

[5] Usama Fayyad,"A Tutorial on Support Vector Machines for,PatternRecognition",burges@lucent.com,Bell,Laboratories ,Kluwer Academic Publishers,2000

[6] Richard A.Johnson,"Applied Multivariate Statistical Analysis",ISBN978-7-302-16518-7.pp.192-200,Jau,2008

[7] Jun-Ying Zhang,"Kernel PCA Feature Extraction for Improvingperformace Optimizition of Kernel"Mathmatics and Phhysics, Electronic Science and Technology of Xi'an University,2008

[8] Yi-Na Wang," Estimates based on Non-Gaussian Kernel PCA Optimization and the Application of Knernel", Electronic Science and Technology of Xi'an University,2006

[9] Bennett,"Support Vector Machines: Hype or Hallelujah?", Mathscience Department Rensselaer Polytechnic Institute Troy, NY 12180,2000

[10] Sebastian Mika,"Kernel PCA and De-Noising in Feature Spaces", GMD FIRST, Rudower Chaussee 5, 12489 Berlin, Germany,2005

[11] Box,G.E.P.,and D.R.COX,"An Analysis of Transformations", Journal of the Royal Statistical Society ,26,no. 2,211-252

[12]Hernandez, F., and R.A.Johnson. "The Large-Sample Behavior of Transformations to Normality." Journal of the American Statistical Association, 75, no. 372,855-861

An Automated Derivation of Łukasiewicz's *CN* Sentential Calculus from Church's P_2

Jack K. Horner
P. O. Box 266
Los Alamos, New Mexico 87544 USA

Abstract

Two logics are implicationally equivalent if the axioms and inference rules of each imply the axioms of the other. Characterizing the implicational equivalences of various formulations of the sentential calculi is foundational to the study of logic. Using an automated deduction system, I show that Łukasiewicz's CN sentential calculus can be derived from Church's P_2; the proof appears to be novel.

Keywords: propositional logic, automated deduction, sentential calculus

1.0 Introduction

Two logics are implicationally equivalent if the axioms and inference rules of each imply the axioms of the other. Characterizing the implicational equivalences of various formulations of the sentential calculi is foundational to the study of logic ([1],[3]-[7],[9]-[10],[12]-[15]).

"CN", the formulation of the sentential calculus in [1], is among the most austere: its vocabulary contains only two logical connectives (C, and N) and sentence variables (*p*, *q*, *r*, ...). It has two inference rules (condensed detachment and substitution), and three axioms.

In CN, any expression of the form *Cxy* or *Nz*, where *x*, *y*, and *z* are sentences, is a sentence. *Cpq* is interpreted as "sentence *p* implies sentence *q*"; *Np* is interpreted as "not-*p*". C and N are right-associative; N has higher associative precedence than C. For example,

CCqrCpNr

translates to the more common "arrow-and-parenthesis" notation as

$$(q \rightarrow r) \rightarrow (p \rightarrow \sim r)$$

where "$\rightarrow$" designates "implies" and "~" designates "not".

The axioms of CN in [1] are:

CN1. *CCpqCCqrCpr*
CN2. *CCNppp*
CN3. *CpCNpq*

Cast in CN notation, the axioms of the Church's P_2 sentential calculus (P2, [12]) are

P2-202. *CpCqp*
P2-203. *CCsCpqCCspCsq*
P2-204. *CCNpNqCqp*

The main result of this paper is that [12] implies [1].

2.0 Method

To show that [12] implies [1], the *prover9* ([2]) script shown in Figure 1 was executed under on a Dell Inspiron 545 with an Intel Core2 Quad CPU Q8200 @ 2.33 GHz and 8.00 GB RAM, running under the *Windows Vista Home Premium (SP2)/Cygwin* operating environment.

```
set(hyper_resolution).

formulas(usable).
P ( i(x, i(y,x)) )                          # label("P2-202").
P ( i(i(z, i(x,y)), i(i(z,x), i(z,y))) )    # label("P2-203").
P ( i(i(-x,-y), i(y,x)) )                   # label("P2-204").
end_of_list.

formulas(sos).
-P(i(x,y)) | -P(x) | P(y)           # label("InfConDet").
end_of_list.

formulas(goals).
P( i(i(x,y), i(i(y,z), i(x,z))) )       # label("AxCN1").
P( i(i(-x,x), x) )                      # label("AxCN2").
P( i(x, i(-x,y)) )                      # label("AxCN3").
end_of_list.
```

Figure 1. The *prover9* script used to show that P2 implies CN. The implementation of condensed detachment is the formula in the "sos" list; substitution is derived from *prover9*'s hyperresolution rule (introduced in the "set" command at the top of the script). Details of *prover9*'s syntax and semantics can be found in [2].

3.0 Results

Figure 2 shows that P2 implies CN.

```
============================== PROOF =================================

% Proof 1 at 0.11 (+ 0.03) seconds: "AxCN2".
% Length of proof is 18.
% Level of proof is 6.
% Maximum clause weight is 20.
% Given clauses 97.

2 P(i(i(-x,x),x)) # label("AxCN2") # label(non_clause) # label(goal).
[goal].
4 P(i(x,i(y,x))) # label("P2-202").  [assumption].
5 P(i(i(x,i(y,z)),i(i(x,y),i(x,z)))) # label("P2-203").  [assumption].
6 P(i(i(-x,-y),i(y,x))) # label("P2-204").  [assumption].
7 -P(i(x,y)) | -P(x) | P(y) # label("InfConDet").  [assumption].
9 -P(i(i(-c4,c4),c4)) # label("AxCN2") # answer("AxCN2").  [deny(2)].
11 P(i(i(i(-x,-y),y),i(i(-x,-y),x))).  [hyper(7,a,5,a,b,6,a)].
12 P(i(i(i(x,i(y,z)),i(x,y)),i(i(x,i(y,z)),i(x,z)))).
[hyper(7,a,5,a,b,5,a)].
14 P(i(x,i(i(-y,-z),i(z,y)))).  [hyper(7,a,4,a,b,6,a)].
16 P(i(x,i(y,i(z,y)))).  [hyper(7,a,4,a,b,4,a)].
18 P(i(x,i(i(i(-y,-z),z),i(i(-y,-z),y)))).  [hyper(7,a,4,a,b,11,a)].
33 P(i(i(x,i(i(y,x),z)),i(x,z))).  [hyper(7,a,12,a,b,16,a)].
152 P(i(x,i(i(-y,-x),y))).  [hyper(7,a,33,a,b,18,a)].
154 P(i(-x,i(x,y))).  [hyper(7,a,33,a,b,14,a)].
158 P(i(i(-x,x),i(-x,y))).  [hyper(7,a,5,a,b,154,a)].
185 P(i(i(x,i(-y,-x)),i(x,y))).  [hyper(7,a,5,a,b,152,a)].
1009 P(i(i(-x,x),x)).  [hyper(7,a,185,a,b,158,a)].
1010 $F # answer("AxCN2").  [resolve(1009,a,9,a)].

============================== end of proof ==========================

============================== PROOF =================================

% Proof 2 at 0.11 (+ 0.03) seconds: "AxCN3".
% Length of proof is 25.
% Level of proof is 9.
% Maximum clause weight is 20.
% Given clauses 101.

3 P(i(x,i(-x,y))) # label("AxCN3") # label(non_clause) # label(goal).
[goal].
4 P(i(x,i(y,x))) # label("P2-202").  [assumption].
5 P(i(i(x,i(y,z)),i(i(x,y),i(x,z)))) # label("P2-203").  [assumption].
6 P(i(i(-x,-y),i(y,x))) # label("P2-204").  [assumption].
7 -P(i(x,y)) | -P(x) | P(y) # label("InfConDet").  [assumption].
10 -P(i(c5,i(-c5,c6))) # label("AxCN3") # answer("AxCN3").  [deny(3)].
11 P(i(i(i(-x,-y),y),i(i(-x,-y),x))).  [hyper(7,a,5,a,b,6,a)].
12 P(i(i(i(x,i(y,z)),i(x,y)),i(i(x,i(y,z)),i(x,z)))).
[hyper(7,a,5,a,b,5,a)].
13 P(i(i(x,y),i(x,x))).  [hyper(7,a,5,a,b,4,a)].
14 P(i(x,i(i(-y,-z),i(z,y)))).  [hyper(7,a,4,a,b,6,a)].
16 P(i(x,i(y,i(z,y)))).  [hyper(7,a,4,a,b,4,a)].
18 P(i(x,i(i(i(-y,-z),z),i(i(-y,-z),y)))).  [hyper(7,a,4,a,b,11,a)].
```

```
22 P(i(x,x)).  [hyper(7,a,13,a,b,4,a)].
23 P(i(i(i(x,y),x),i(i(x,y),y))).  [hyper(7,a,5,a,b,22,a)].
33 P(i(i(x,i(i(y,x),z)),i(x,z))).  [hyper(7,a,12,a,b,16,a)].
152 P(i(x,i(i(-y,-x),y))).  [hyper(7,a,33,a,b,18,a)].
154 P(i(-x,i(x,y))).  [hyper(7,a,33,a,b,14,a)].
159 P(i(x,i(-y,i(y,z)))).  [hyper(7,a,4,a,b,154,a)].
165 P(i(i(i(-x,i(x,y)),z),z)).  [hyper(7,a,23,a,b,159,a)].
168 P(i(i(x,-y),i(x,i(y,z)))).  [hyper(7,a,5,a,b,159,a)].
185 P(i(i(x,i(-y,-x)),i(x,y))).  [hyper(7,a,5,a,b,152,a)].
996 P(i(--x,x)).  [hyper(7,a,165,a,b,185,a)].
1029 P(i(x,--x)).  [hyper(7,a,6,a,b,996,a)].
1050 P(i(x,i(-x,y))).  [hyper(7,a,168,a,b,1029,a)].
1051 $F # answer("AxCN3").  [resolve(1050,a,10,a)].

============================== end of proof ==========================

============================== PROOF =================================

% Proof 3 at 4.93 (+ 0.53) seconds: "AxCN1".
% Length of proof is 18.
% Level of proof is 8.
% Maximum clause weight is 24.
% Given clauses 1191.

1 P(i(i(x,y),i(i(y,z),i(x,z)))) # label("AxCN1") # label(non_clause) #
label(goal).  [goal].
4 P(i(x,i(y,x))) # label("P2-202").  [assumption].
5 P(i(i(x,i(y,z)),i(i(x,y),i(x,z)))) # label("P2-203").  [assumption].
7 -P(i(x,y)) | -P(x) | P(y) # label("InfConDet").  [assumption].
8 -P(i(i(c1,c2),i(i(c2,c3),i(c1,c3)))) # label("AxCN1") #
answer("AxCN1").  [deny(1)].
12 P(i(i(i(x,i(y,z)),i(x,y)),i(i(x,i(y,z)),i(x,z)))).
[hyper(7,a,5,a,b,5,a)].
15 P(i(x,i(i(y,i(z,u)),i(i(y,z),i(y,u))))).  [hyper(7,a,4,a,b,5,a)].
16 P(i(x,i(y,i(z,y)))).  [hyper(7,a,4,a,b,4,a)].
29 P(i(x,i(y,i(z,i(u,z))))).  [hyper(7,a,4,a,b,16,a)].
33 P(i(i(x,i(i(y,x),z)),i(x,z))).  [hyper(7,a,12,a,b,16,a)].
65 P(i(i(x,i(i(y,i(z,y)),u)),i(x,u))).  [hyper(7,a,12,a,b,29,a)].
89 P(i(i(i(x,i(y,z)),i(i(i(x,y),i(x,z)),u)),i(i(x,i(y,z)),u))).
[hyper(7,a,12,a,b,15,a)].
153 P(i(i(x,y),i(i(z,x),i(z,y)))).  [hyper(7,a,33,a,b,15,a)].
2531 P(i(i(i(x,y),z),i(y,z))).  [hyper(7,a,65,a,b,153,a)].
4557 P(i(x,i(i(i(y,z),u),i(z,u)))).  [hyper(7,a,4,a,b,2531,a)].
20399 P(i(i(x,i(y,z)),i(y,i(x,z)))).  [hyper(7,a,89,a,b,4557,a)].
23067 P(i(i(x,y),i(i(y,z),i(x,z)))).  [hyper(7,a,20399,a,b,153,a)].
23068 $F # answer("AxCN1").  [resolve(23067,a,8,a)].

============================== end of proof ==========================
```

Figure 2. Summary of a *prover9* ([2]) proof showing that P2 ([12]) implies CN ([1]).

The total time to complete the proofs shown in Figure 2 was ~6 seconds on the platform described in Section 2.0.

4.0 Conclusions and discussion

Section 3 demonstrates that P_2 implies CN. A companion paper ([16]) proves CN implies P_2. The proof in Figure 2 appears to be novel.

5.0 References

[1] Łukasiewicz J. *Elements of Mathematical Logic*. Second Edition (1958). Trans. by Wojtasiewicz O. Pergamon Press. 1963.

[2] McCune WW. *prover9 and mace4.* http://www.cs.unm.edu/~mccune/prover9/. 2009.

[3] Aristotle. *Prior Analytics*. Trans. by A. J. Jenkinson. In Aristotle. *The Basic Works of Aristotle.* Ed. by R. McKeon. Random House. 1941. pp. 62-107.

[4] Aristotle. *Posterior Analytics.* Trans. by G. R. G. Mure. In Aristotle. *The Basic Works of Aristotle.* Ed. by R. McKeon. Random House. 1941. pp. 108-186.

[5] Tarski A. *Introduction to Logic*. Trans. by O. Helmer. Dover. 1941.

[6] Hempel C. Studies in the logic of explanation. In Hempel C. *Aspects of Scientific Explanation and Other Essays in the Philosophy of Science.* Free Press. 1965. pp. 245-290.

[7] Quine WVO. *Philosophy of Logic*. Second Edition. Harvard. 1986.

[8] Chang CC and Keisler HJ. *Model Theory*. North-Holland. 1990.

[9] Russell B and Whitehead AN. *Principia Mathematica*. Volume I (1910). Merchant Books. 2009.

[10] Frege G. Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens. Halle. 1879. Translated in van Heijenoort J. Begriffsschrift, a formula language, modeled upon that of arithmetic, for pure thought. *From Frege to Gödel: A Source Book in Mathematical Logic, 1879-1931*. Harvard. 1967. pp. 3-82.

[11] Horn A. On sentences which are true of direct unions of algebras. *Journal of Symbolic Logic* 16 (1951), 14–21.

[12] Church A. *Introduction to Mathematical Logic*. Volume I. Princeton. 1956.

[13] Birkhoff G and von Neumann J. The logic of quantum mechanics. *Annals of Mathematics* 37 (1936), 823-843.

[14] Kant I. *Kant's Introduction to Logic* (1800). Trans. by Abbott TK. Greenwood Press. 1963.

[15] Cohen MR and Nagel E. *An Introduction to Logic and Scientific Method.* Harcourt, Brace, and Company. 1934.

[16] Horner JK. An automated derivation of Church's P_2 sentential calculus from Łukasiewicz's CN. *Proceedings of the 2011 International Conference on Artificial Intelligence.* CSREA Press. Forthcoming.

Improving Network Intrusion Detection with Growing Hierarchical Self-Organizing Maps

Andres Ortiz[1], **Julio Ortega**[2], **Alberto Prieto**[2], **Antonio F. Díaz**[2]
[1] Communications Engineering Department. University of Malaga.
29004 Malaga, Spain
[2] Department of Computer Architecture and Technology. University of Granada.
18060 Granada, Spain

Abstract - *Nowadays, the growth of the computer networks and the expansion of the Internet have made the security to be a critical issue. In fact, many proposals for Intrusion Detection/Prevention Systems (IDS/IPS) have been proposed. These proposals try to avoid that corrupt or anomalous traffic reaches the user application or the operating system. Nevertheless, most of the IDS/IPS proposals only distinguish between normal traffic and anomalous traffic that can be suspected to be a potential attack. In this paper, we present a IDS/IPS approach based on Growing Hierarchical Self-Organizing Maps (GHSOM) which can not only differentiate between normal and anomalous traffic but also identify different known attacks. The proposed system has been trained and tested using the well-known DARPA/NSL-KDD datasets and the results obtained are promising since we can detect over 99,4% of the normal traffic and over 99,2 % of attacker traffic. Moreover, the system can be trained on-line by using the probability labeling method presented on this paper.*

Keywords: IDS, IPS, Attack Classification, Self-Organizing Maps, Growing Self-Organizing Maps, SOM relabeling, clustering.

1 Introduction

The interest in computer network security has increased in recent years, as the trend to on-line services available through the Internet have exposed a lot of sensitive information to intruders and attackers [1]. Although there are several methods to protect the information, there is not any infallible encryption method and the encryption/decryption process can impose a high overhead in high speed networks that use the TCP/IP protocol stack. On the other hand, the complexity of the newer attacks make necessary the use of elaborate techniques such as pattern classification or artificial intelligence techniques for successfully detecting an attack or just to differentiate among normal and abnormal traffic. IDS and IPS systems are active systems which implement a protection by continuously monitoring the network. They calculate some traffic features to be able to classify it, to detect abnormal behaviors and to react according to some predefined rules. There are two design approaches to IDS/IPS systems [2]. The first one consists on looking for patterns corresponding with known signatures of intrusions. The second one searches for abnormal patterns by more complex features, to discover not only intrusions but also potential intrusions.

Several neural and machine learning techniques have been used for implementing IDS/IPS systems [2-6]. In [2], perceptron-like neural networks are used for traffic classification as well as fuzzy classifiers to improve the decision step. In [3, 4], Self-Organizing Maps (SOMs) [5] are applied to implement unsupervised clustering of the data instances in order to classify the traffic anomalies according to several known attack. In order to improve the classification task either by distinguishing the normal traffic from anomalies or by classifying the different attacks with high accuracy, hierarchical SOM have been used in works. Although the proposal on [3] tries to overcome some of the difficulties on the static structure of classic SOMs splitting the SOM into three smaller SOMs, the size of these maps is still static.

An alternative to avoid the limitations of the classical SOM is the Growing Hierarchical SOM (GHSOM) [7]. It presents a dynamic hierarchical structure composed of several layers, with several SOMs in each layer. The number of SOMs in each layer and their respective sizes are determined in the GHSOM training process. In this paper, GHSOM is used for both, anomaly detection and attack classification. Moreover, a probability-based mechanism applied to the previously trained structure is used to label the units when the GHSOM is applied to new data instances.

After this introduction, the remainder of this paper is organized as follows. Section 2 shows the features of the NSL-KDD dataset that, as in most of the IDS/IPS works, has been used to evaluate our system. Section 3 describes our proposals for data preprocessing and for applying GHSOM to attack classification. In Section 4, the results obtained are shown, and finally, Section 5 provides the conclusions of the paper and our future work

2 Attack Classification with GHSOM

The first step in IDS and IPS implementation is the extraction of some significant features from the captured traffic. The features on the NSL-KDD dataset [8] (the benchmark set we have used in this work) are split into three classes: basic features, content-based features, and traffic features. The main reason for having these three groups of features is that detecting and identifying some attacks requires the use of more than just one feature class. For instance, time-based features are necessary to detect some attacks, as some statistics should be calculated over a certain time period. Thus, the first step we perform consists of parsing the dataset in order to extract the features from the text files and to build the vectors which comprise the feature space. This feature space is composed of vectors belonging to R^{41} which contains the connection traffic features. Examples of these features are the duration of the connection, the protocol type, or the number of *user-to-root* attempts. Nevertheless, the feature selection for network-based IDS is not straightforward. This way, there are works [10, 11, 12] which use multivariate techniques such as PCA [10] or LDA [11]. The use of multivariate techniques try to obtain the components with the highest variability, supposing the rest are unimportant or just noise. Although good results are given in [10, 11, 12] other works have shown that the full set of features outperforms feature selection with PCA/LDA [13]. Moreover, feature reduction has to be applied to each input vector since the most discriminate component depends on the specific attack. Thus, in [18] a SOM classifier for attack detection is presented without reducing the feature space. However, the authors only use a selection of 28 features from the 41 available in the NSL-KDD dataset, and the influence of each feature is figured out through the *U-Matrix* of each component belonging to the input (feature) space. Instead, in this paper we propose the use of a dynamic structure such as GHSOM and the full set of features.

2.1 SOMs and GHSOMs

SOM is a very useful tool for discovering structures and similarities in high dimensional data, organizing the information and visualizing it in a 2D or 3D way. Detailed descriptions of SOM can be found elsewhere [5]. Nevertheless, SOM is not able to figure out the inherent hierarchical structure of data [7], and, at the same time, the performance of SOM depends on the size of the map which has to be set in advance. Thus, GHSOM [7] is a hierarchical and non-fixed structure developed to overcome the main limitations of classical SOM. The structure of GHSOM consists of multiple layers composed by several independent SOMs. Hence, during the learning process, the number of the SOMs on each layer and the size of each SOM are determined by minimizing the quantization error. Thus an adaptive growing process is accomplished by using two parameters τ_1 and τ_2 that respectively control the breadth of each map (horizontal direction) and the growing of the hierarchy (vertical direction). Therefore, these two parameters are the only parameters that have to be set in advance.

In order to determine how far the GHSOM grows [7], the quantization error of each unit is calculated according to the Equation 1, where Ci is the set of input vectors mapped to the i-th unit, xj is the j-th input vector belonging to Ci, and ωi is the weight associated to the i-th unit.

$$qe_i = \sum_{x_j \in C_i} \| w_i - x_j \| \quad (1)$$

Initially, all the input vectors belong to C0 . This means all the inputs are used to compute the initial quantization error, qe0. Then, the quantization errors qei for each neuron are calculated. Thus, whenever qei<τ_2< qe0, the i-th neuron is expanded in a new map on the next level of the hierarchy. Each new map is trained as an independent SOM, and the calculation of its BMU (Best Matching Unit) is done by using the Euclidean distance. Once the new map is trained, the quantization error of each neuron on this map is computed. Then, the mean quantization error MQEm of the new map can be determined. Whenever MQEm<τ_1·qeu (qeu is the quantization error of the unit u on the upper layer), the map stops growing. The process has been schematized in Figure 1 and it is explained with detail in [9].

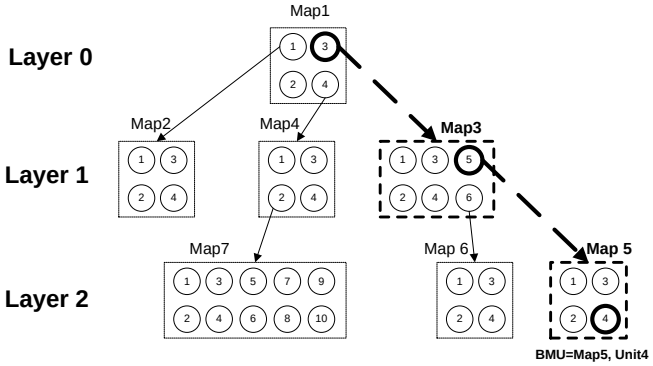


Figure 1. BMU calculation on the GHSOM hierarchy

To apply GHSOM to IDS/IPS, it is necessary to encode some qualitative features included in the NSL-KDD dataset due to the numeric nature of the GHSOM input vectors. For example, this is the case of protocol, service and flag features. The encoding of these features is performed by assigning numeric values in order t keep a high enough distance among them for the effectiveness of the SOM classifier. Specifically, the number 1 and other prime numbers with distances higher than 6 among them have been chosen. This coding has been chosen in order to increase the distance among different features. For example, in the component protocol in the input vectors, TCP is encoded as 1, UDP as 7, and ICMP as 17. In order to avoid that some features have more influence than others, the input vectors have been normalized by subtracting

the mean and dividing by the standard deviation (zero mean and unity variance). Thus, each dimension takes a value between 0 and 1, and the feature space belongs to R^{41}.

2.2 BMU calculation for GHSOM

In order to calculate the BMU in the GHSOM, we have to go through all the hierarchy to determine the winning unit and the map to which it belongs. Thus, an iterative algorithm has been developed as shown in Figure 1, where an example of BMU calculation on a three-level GHSOM hierarchy is considered. After computing the distances between an input pattern and the weight vectors of the map in layer 0, the minimum of these distances is determined. Once, the winning neuron on map 1 is found, since other map could be grown from this winning neuron, we have to check whether the wining neuron is a parent unit. This can be accomplished with the parent vectors resulting from the GHSOM training process. If a new map arose from the wining neuron, the BMU on this new map is calculated. This process is repeated until a BMU with no growing map is found. Thus, the BMU in the GHSOM is identified inside a map in a layer of the hierarchy (for example, map 5 and unit 4).

2.3 GHSOM training and relabeling

The GHSOM structure has been trained by using 10% of the training samples provided by the NSL-KDD dataset. Then, the system has been tested by using the rest of the training patterns. As it is commented in Section 2.2, we have used the full set of features. After several tests using different values for τ_1 (to control the breadth of the map) and τ_2 (to control the depth), we have selected τ_1=0.6 and $\tau_2=10^{-5}$. This could make the GHSOM to grow more than necessary, leaving some of the units unlabeled. Thus, the number of neurons on the output map surrounding the winning neuron for training data is increased. When an input pattern similar to one of the training patterns is presented to the GHSOM, the wining neuron can be labeled or unlabeled. If the wining neuron is unlabeled, a probability-based scheme is used in order to determine the label of that neuron. More specifically, the wining neuron is labeled (or relabeled) with the more repeated label in its neighborhood by using a probability calculated according to (2).

$$P_u = \frac{M_{\sigma(u,\varepsilon)}(u)}{n} \quad (2)$$

In this expression, P_u is the probability for the winning unit u to be successfully relabeled, where $M_{\sigma(u,\varepsilon)}(u)$ is the label that appears more frequently in the Gaussian neighborhood $\sigma(u,\varepsilon)$, of the winning neuron, u , and n is the number of neurons belonging to the neighborhood of u, $\sigma(u,\varepsilon)$. In this equation, parameter ε noted the width of the neighborhood of the winning neuron.

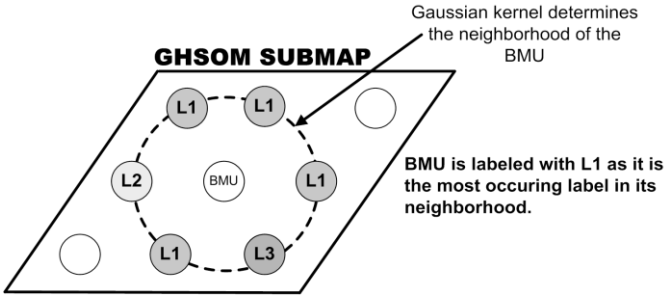


Figure 2. Example of the relabeling process (white units are unlabeled).

In Figure 2, an example of the relabeling process is shown. In this Figure, the BMU that has not been initially labeled, is labeled by using ε=1 to establish its neighborhood. In this neighborhood, we found four units labeled as L1, one unit labeled as L2 and one unit labeled as L3. Then, P_u, the probability for successful relabeling for this BMU is 4/6=0.66 (66%) ($M_{\sigma(u,\varepsilon)}(u)$) =4, n=6). this BMU is 4/6=0.66 (66%) ($M_{\sigma(u,\varepsilon)}(u)$) =4, n=6).

3 Experimental results

In this section, we present the experimental results obtained with the NSL-KDD dataset [9] and the GHSOM classifier described in Section 2. In Figure 3, the detection success for each type of attack included in the NSL-KDD dataset is shown. As this figure shows, the probability-based labeling process performed with new data (black bar in Figure 2) increases the detection success for most attacks. Moreover some attacks such as *multihop*, are not detected before the unit relabeling process.

In order to show the effectiveness of the relabeling process due to the associated probability, the ROC (receiver operating characteristic) curves are shown in Figure 4. They constitute an effective alternative to evaluate the performance of a classifier.

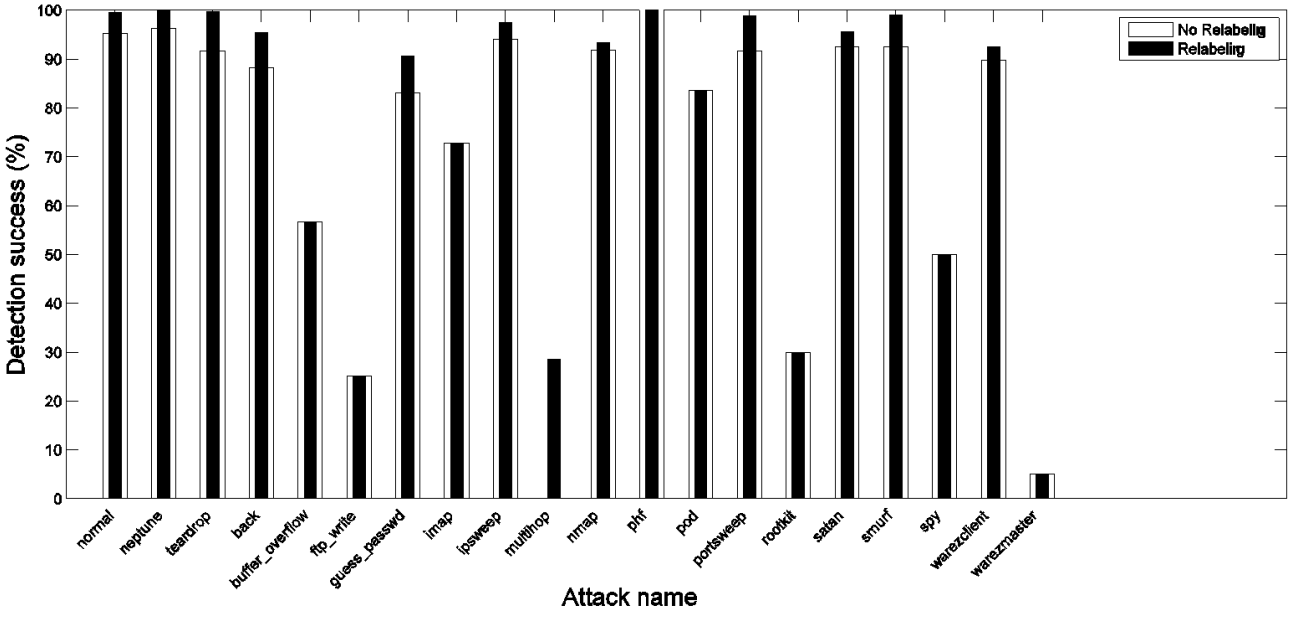
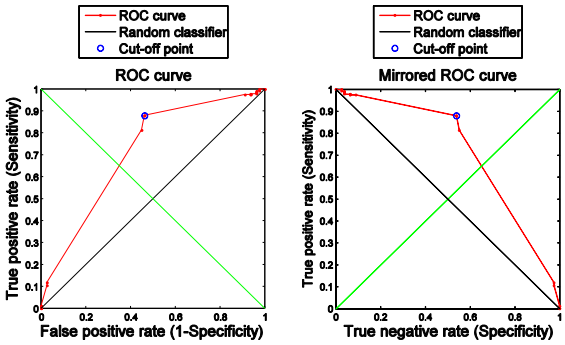


Figure 3. Detection success with (black bar) and without (white bar) unit relabeling

Figure 4.a shows the ROC curve (false positive rate) and Figure 5 the mirrored ROC curve (true negative rate). Regarding a measure of performance derived from these curves, we computed the *Area Under ROC Curve* (AUC). The use of AUC makes the interpretation of the results from the ROC curve easier. Thus, a perfect classifier will provide an AUC=1.0 whereas in a random classifier AUC=0.5. In the graphs of Figure 4, the cut point determines the best performance the classifier can provide. The AUC computed from our ROC curves is 0.71. Hence, the AUC is statistically grater than 0.5 which denotes a fair behavior of the relabeling process.



(a) (b)

Figure 4. ROC curves for the relabeling process.

In Figure 5, we present the detection success rate per attack type. As can be seen, in all cases, the relabeling method increases the classification performance as well as the detection success rate. Regarding to *User to Root* (U2R) attacks, the performance is worse than that obtained for other attacks. Nevertheless, the number of U2R training patterns on the NSL-KDD is significantly less than for other attacks [14].

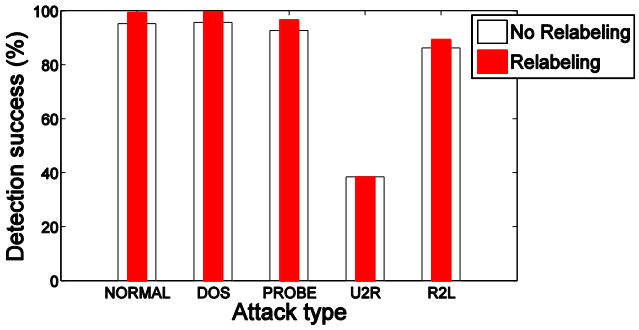


Figure 5. Detection Success rate for different types of attac.

Moreover, Figure 6 summarizes the performance of our proposal when detecting normal/abnormal traffic. As shown in this figure, 99.6% of normal traffic patterns and 99.2% of the attack patterns have been correctly classified.

In Table 1, testing results for previous proposed IDSs based on SOM and GHSOM are extracted from [14]. As it is shown in this table, our GHSOM with relabeling probabilies (RL-GHSOM in Table 1) reaches a high rate of detected attacks and clearly outperforms the false positive rate provided by other similar proposals.

Table 1. Basic features of individual TCP connections

IDS implementation	Detected attacks (%)	False Positive (%)
RL-GHSOM	99.68	0.02
GHSOM	99.99	3.72
K-Map	99.63	0.34
SOM	97.31	0.04

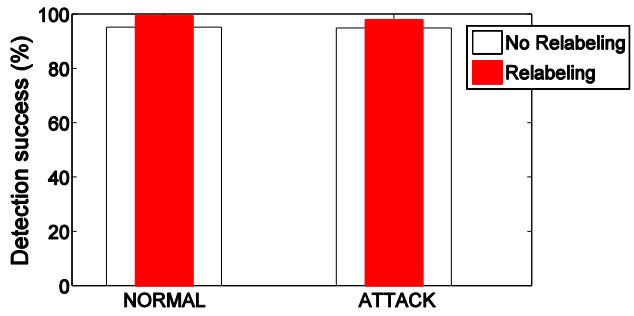


Figure 6. Detection Success rate for normal/attack traffic. Detection success with unit relabeling (black bar) and without unit relabeling (white bar).

4 Conclusions and future directions

In this paper we present a network intrusion prevention approach that takes advantage of the discriminating properties of the GHSOM. Moreover, instead of applying any feature selection technique over the dataset, the full set of data features has been used. This circumstance has required to let the GHSOM grow more than it should be necessary and to devise a labeling (or relabeling) process for the BMUs that uses a probability for relabeling success. Acceptable results have been obtained from an analysis of the effectiveness of the relabeling process which has been done by using the ROC curves. The results obtained for the proposed IPS are promising, since it can detect 99.6% of the normal traffic patterns and 99.2% of the abnormal ones.

As future work, we will consider a real-time implementation of the IPS. This could be feasible as we avoid the need for principal component analysis. With such kind of implementation, as normal and abnormal behaviors could be accurately detected on line, it would be possible to perform some complementary activities, such as IP blocking, in real time in order to improve the quality of the network intrusion prevention.

Acknowledgments. This work was supported by project SAF2010-20558 (Ministerio de Educación, Spain).

References

[1] Ghosh, J., Wanken, J., Charron, F.: Detecting anomalous and unknown intrusions against programs. Proceedings of the Annual Computer Security Applications Conference, 1998.

[2] Hoffman, A., Schimitz, C., Sick, B.: Intrussion Detection in Computer networks with Neural and Fuzzy classifiers. International Conference on Artificial Neural Networks, ICANN 2003.

[3] Lichodzijewski, P., Zincir-Heywood, N., Heywood, M.: Host Based Intrusion Detection Using Self-Organizing Maps. Proceedings of the IEEE International Joint Conference on Neural Networks. 2002.

[4] Zhang, C., Jiang, J., Kamel, M.: Intrusion Detection using hierarchical neural networks. Pattern Recognition Letters, issue 26 (2005), pp. 779-791.

[5] Kohonen T. Self-Organizing Maps. Springer, third edition, 2001.

[6] Fisch, D., Hofmann, A., Sick, B.: On the versatility of radial basis function neural networks: A case study in the field of intrusion detection. Inf. Sci. 180(12): 2421-2439 (2010),

[7] Rauber, A., Merkl, D., Dittenbach, M.: The Growing Hierarchical Self-Organizing Map: Exploraroy Analysis of High-Dimensional Data. IEEE Transactions on Neural Network, Vol. 13, nº6. 2002.

[8] Oh, H., Doh, I., Chae, K.: Attack Classification based on data mining technique and its application for reliable medical sensor communication. International Journal Of Science and Applications. Vol. 6, nº3, pp. 20-32. 2009.

[9] The NSL-KDD dataset. http://iscx.ca/NSL-KDD/

[10] Lakhina, S., Joseph, S., Verma, B.: Feature Reduction using Principal Component Analysis for Effective Anomaly-Based Intrusion Detection on NSL-KDD. International Journal on Engineering Science and Technology. Vol. 2(6), 2010, pp. 1790-1799.

[10] Datti, R., Verma, B.: Feature Reduction for Intrusion Detection Using Linear Discriminant Analysis. International Journal on Engineering Science and Technology. Vol. 2(4), 2010, pp. 1072-1078.

[12] Zargar, G.R., Kabiri, P.: Selection of Effective Network Parameters in Attacks for Intrussion Detection. IEEE International Conference on Data Mining. 2010.

[13] Mukkamala, S., Sung, A.H.: Feature Ranking and Selection for Intrusion Detection Systems Using Support Vector Machines. Proceedings of the Second Digital Forensic Research Workshop. 2002.

[14] Palomo, E.J., Domínguez, E., Luque, R.M., Muñoz, J.: Network Security Using Growing Hierarchical Self-Organizing Maps. International Conference on Adaptive and Natural Computing Algorithms, ICANNGA 2009.

Evaluation of Illuminance Provided by the Intelligent Lighting System in Actual Office

Mitsunori MIKI
Department of Science and Engineering
Doshisha University
Kyoto, Japan
Email: mmiki@mail.doshisha.ac.jp

Yoshihiro KASAHARA
Graduate School of Engineering
Doshisha University
Kyoto, Japan
Email: ykasahara@mikilab.doshisha.ac.jp

Tomoyuki HIROYASU
Department of Life and Medical Sciences
Doshisha University
Kyoto, Japan
Email: tomo@is.doshisha.ac.jp

Masato YOSHIMI
Department of Science and Engineering
Doshisha University
Kyoto, Japan
Email: myoshimi@mail.doshisha.ac.jp

Hirotaka Ito
Graduate School of Engineering
Doshisha University
Kyoto, Japan
Email: hito@mikilab.doshisha.ac.jp

***Abstract*—In our laboratory, lighting control system targeting office environment called Intelligent Lighting System has been proposed. Intelligent Lighting system is not the existing lighting system that provides uniform illuminance, but it can individually provide different illuminance depending on a worker's work content and preference. Presently, prototype Intelligent Lighting Systems are introduced to some office buildings to conduct demonstration experiment for commercialization. Illuminance condition of areas where illuminance sensors were installed is achieved in the demonstration experiment, and it was confirmed that each worker can work comfortably with illuminance they selected. However, illuminance of the entire room provided by Intelligent Lighting System had not been reviewed. Evaluation by illuminance distribution is also important for Intellectual Lighting System since it provides different illuminance to workers. Therefore, this study mentions evaluation of illuminance provided by Intelligent Lighting System in actual office environment by using the system which can measure illuminance distribution.**

***Index Terms*—Optimization, Lighting Control, Illuminance Distribution, Office Environment**

I. Introduction

As electronic control technologies and information processing technologies develop in recent years, intelligence has been incorporated in various devices and systems including electric appliances and automobiles, where the system autonomously controls its movement and management depending on a user or environment and burdens to people are reduced. The intelligent system has the ability to take proper actions by thinking, understanding and making judgments based on its own knowledge obtained from information using a sensor, etc. With the intelligent system, autonomous movements in response to the surrounding environment are possible, user satisfaction is improved, and it is possible to flexibly respond to various environmental changes[1].

On the other hand, improvement of office workers' intelligent productivity, creativity and comfortableness in offices has been focused on in recent years[2], [3]. And it is clarified in the study by Boyce, etc. that to provide illuminance most suitable for execution of work for each individual is effective from the viewpoint of improving the lighting environment[4]. To provide brightness most suitable for execution of work for each individual is easily realized with task and ambient lighting. However, ceiling lighting fixtures which provide even brightness on a floor are common in office buildings in Japan, and it is not easy to adopt task and ambient lighting. Therefore, the lighting control system to provide brightness most suitable for each office worker is necessary by using ceiling lighting fixtures.

Based on the above viewpoints, our laboratory has been conducting research on intelligent lighting system which provides illuminance appropriate for office workers' needs. The intelligent lighting system consists of lighting fixtures equipped with a microprocessor, illuminance sensors and an energy meter connected to the network. Intelligent lighting system allows each worker to set the ideal illuminance on his/her own illuminance sensor. The system can achieve a lighting pattern with the smallest electric power and at the same time satisfy the set illuminance value by using optimization method. A basic experiment revealed that intelligent lighting system presently is able to satisfy a user's ideal illuminance and achieve high rate of energy saving. Verification experiments have been conducted for actual use in offices[5].

The intelligent lighting system uses an illuminance sensor installed at each worker's desk for controlling. Therefore, the illuminance condition is satisfied in an area where an illuminance sensor is installed, and each worker can work comfortably under the brightness s/he selected. However, illuminance had not been examined in areas around where illuminance sensors are installed and where the sensors are not installed. Therefore, our laboratory has developed illuminance distribution measurement system in order to evaluate such illuminance[7].

This study mentions evaluation result of illuminance provided by Intelligent Lighting System in actual office environment by using illuminance distribution measurement system. Evaluation items are; illuminance of areas around where illuminance sensors are installed (work area) and of areas where illuminance sensors are not installed (allies, etc.).

CIE recommends the balance rate (ratio of minimum illuminance against average illuminance) in a work area shall be 0.7 or more, and 0.5 or more around a work area for illuminance evaluation in this work area[8]. Therefore, it is necessary to evaluate illuminance in work areas based on this standard for the intelligent lighting system. On the other hand, locations such as corridors where illuminance sensors are not installed are places not requiring illuminance provision. They may, therefore, be considered to have low illuminance, and need to have illuminance evaluation by illuminance measurement.

II. Intelligent Lighting System

A. Onerview of the Intelligent Lighting System

The intelligent lighting system, as indicated in Fig. 1, is composed of lights equipped with microprocessors, portable illuminance sensors, and electrical power meters, with each element connected via a network.

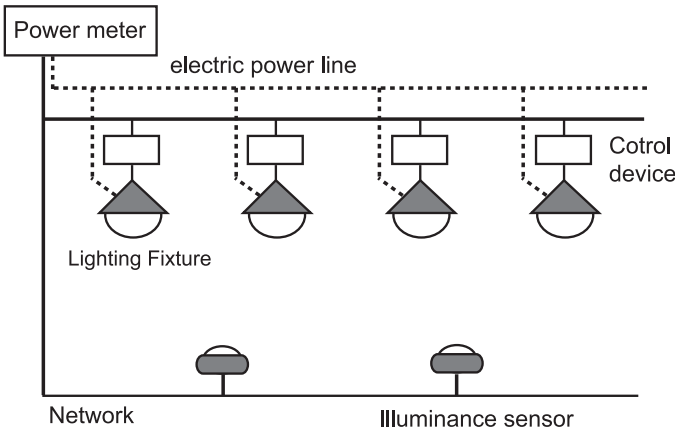


Fig. 1. Configuration of Intelligent Lighting System

Individual users set the illuminance constraint on the illuminance sensors. At this time, each light repeats autonomous changes in luminance to converge to an optimum ligthing pattern. Also, with the intelligent lighting system, positional information for the lights and illuminance sensors is unnecessary. This is because the lights learn the factor of influence to the illuminance sensors, based on illuminance data sent from illuminance sensors. In this fashion, each user's target illuminance can be provided rapidly.

The most significant feature of the intelligent lighting system is that no component exists for integrated control of the whole system; each light is controlled autonomously. For this reason, the system has a high degree of fault tolerance, making it highly reliable even for large-scale offices.

B. Control of the Intelligent Lighting System

In the intelligent lighting system, the algorithm where Simulated Annealing (SA) is improved for lighting control (Adaptive Neighborhood Algorithm using Regression Coefficient: ANA/RC) is used to control luminance intensity for each lighting fixture[9], [10].

SA is the algorithm to obtain the optimal solution by randomly generating the subsequent solution near the present solution, to receive the solution depending on the change in the objective function value as well as on the temperature parameter, and to repeat the transitioning processing. However, using SA is not easy for systems being always necessary to respond to an environmental changes, because SA uses the temperature parameter and cooling method. Then, ANA/RC is proposed. ANA/RC obtains the optimal solution by using a variable neighborhood method without using the temperature parameter that is proposed[9], [10].

It is possible with ANA/RC to provide the target illuminance with minimum power consumption by making luminance intensity for lighting fixtures the design variable and by using the difference between the current illuminance and target illuminance as well as power consumption as objective functions. Furthermore, by learning the influence of each lighting fixture on each illuminance sensor using the regression analysis and by changing the luminance intensity depending on the results, it is possible to promptly change to the optimal luminance intensity. This algorithm is effective to solve the problem which the objective function is near monomodal function and changes in real time.

1) Establish initial parameters including initial luminance intensity.
2) Illuminate each light at the initial luminance intensity.
3) Obtain information on illuminance from each illuminance sensor.
4) Estimate power consumption based on the luminance intensity for each light.
5) Calculate the objective function value in the current luminance intensity.
6) Determine the proper range where the next luminance intensity is generated (proximity or neighborhood) based on the regression coefficient.
7) Randomly generate the next luminance intensity within the neighborhood of (6) and illuminate the lighting fixture at the next luminance intensity.
8) Obtain information on illuminance from each illuminance sensor.
9) Estimate power consumption based on luminance intensity for each light.
10) Calculate the objective function value in the next luminance intensity.
11) Conduct regression analysis based on the amount of change in the luminance intensity for the lighting as well as on the amount of change in illuminance for illuminance sensors.
12) Accept the next luminance intensity if the objective function value turns good. If not, return to the previous luminance intensity.
13) Return to (3).

Through repetition of the above actions, each light learns the factor of influence for each illuminance sensor, each user's target illuminance is provided, and energy consumption is re-

duced. Furthermore, steps (3) to (12) are one search operation of the luminance value, the time required for each search operation is around 2 second. The abovementioned repetitions return to step (3) instead of step (8) (conducting evaluation of objective function once more) in order to allow for any changes in environment. The objective function is indicated in the Equation 1.

$$f = P + w \times \sum_{j=1}^{n} g_j \quad (1)$$

$$P = \sum_{i=1}^{m} L_i$$

$$g_j = \begin{cases} 0 & (Ic_j - It_j) \geq 0 \\ R_j \times (Ic_j - It_j)^2 & (Ic_j - It_j) < 0 \end{cases}$$

$$R_j = \begin{cases} r_j & r_j \geq T \\ 0 & r_j < T \end{cases}$$

n : number of Illuminance sensors, m : number of Lighting fixtures, w : weight, P : electric energy, Ic : Current illminance, It : Target illuminance, L : luminance Intensity, r : regression coefficient, T : Threshold

As indicate in the Equation 1, the objective function f consists of power consumption P and constraint g_j. The difference between the current illuminance and target illuminance is used for the constraint g_j, and a penalty is imposed only if the target illuminance is not achieved. As a result, the objective function value largely increases as the target illuminance goes further than the current illuminance. $R_j = 0$ is multiplied if the regression coefficient is less than the threshold. With this, if the illuminance sensor with a lower regression coefficient does not achieve the target illuminance, the objective function value does not increase. Therefore, objects for optimization are successfully limited to illuminance sensors to which the lighting gives a strong influence. Furthermore, the weight w value is multiplied for constraint g_j, and it is possible to switch whether or not to prioritize the convergence to the target illuminance over minimization of power consumption by setting the weight w value.

C. The Intelligent Lighting System in Actual Office Environment

Presently, the intelligent lighting system is under a verification experiment in some offices in the aim of commercial use. Introduction offices are shown below.

- Kokuyo Co., Ltd. Eco Liveoffice Shinagawa (Tokyo showroom, Minato-ku, Tokyo)
- Mitsubishi Estate Co., Ltd. Area Planning Office (Otemachi Building, Chiyoda-ku,Tokyo)
- Mitsubishi Estate Co., Ltd. Management Building Office (Otemachi Building, Chiyoda-ku,Tokyo)
- Mitsubishi Estate Co., Ltd. Ecozzeria (Shin-Marunouchi Building, Chiyoda-ku, Tokyo)
- Mori Building Co., Ltd. headquarters (Roppongi Hills Mori Tower, Minato-ku, Tokyo)
- Mitsubishi Electric Co. headquarters (Tokyo Building, Chiyoda-ku,Tokyo)

"Ecozzeria", a demonstration office for next-generation low-carbon type technology in Shin-Marunouchi Building (Chiyoda-ku, Tokyo) owned by Mitsubishi Estate Co.,Ltd. is one of these buildings. 11 office workers are working in a floorage of 10.1 m $\times$ 7.2 m in Ecozzeria. Each worker is given a fixed seat, and each desk is installed with 1 illuminance sensor each. A meeting space built in the office is installed with 2 illuminance sensors. Fig.2 displays an example of illuminance convergence in Ecozzeria.

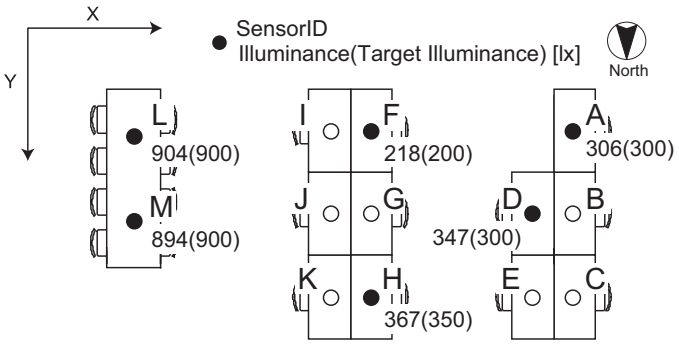


Fig. 2. An example of illuminance convergence in Ecozzeria

In Fig.2, a place displaying a white illuminance sensor implies that a worker is away from the desk. Additionally, since Ecozzeria has windows on the east side, illuminance tends to be higher due to an influence of outside light. The intelligent lighting system, however, allows us to satisfy required illuminance, and as well achieve energy saving by keeps down lighting luminosity of lighting equipments. Fig.2 indicates that the intelligent lighting system satisfies a worker's required illuminance in actual office environment where each illuminance sensor is installed.

However, illuminance in an area surrounding a place where an illuminance sensor is installed had not been reviewed in the past. In this study, therefore, we conduct verification of illuminance in each worker's working area and in a place where illuminance sensor is not installed.

III. Evaluation of Intelligent Lighting System in Actual Office Environment

A. Illuminance Distribution Measurement System

Illuminance distribution in an office is often acquired by measurement using multiple illuminance sensors or calculator simulation, and calculation accuracy verification examples have been also reported[11]. Nonetheless, there are so few illuminance sensors in measurement using multiple illuminance sensors, and measurement by computer simulation cannot deal with an influence of outside light changing by season and weather, an influence of direct sunlight changing in a short time, or deterioration or stain differing by lighting equipment so that accurate illuminance evaluation is not easy. Therefore, we structured a system which visualizes the illuminance data obtained from more than 200 illuminance sensors as real-time illuminance distribution.

Illuminance distribution measurement system consists of multiple illuminance measurement module. These illuminance measurement module are composed of 6 illuminance sensors and an 1 sensor data transmission device. Fig.3-(a) shows the composition of illuminance distribution measurement system and Fig.3-(b) shows actual structure of illuminance distribution measurement system.

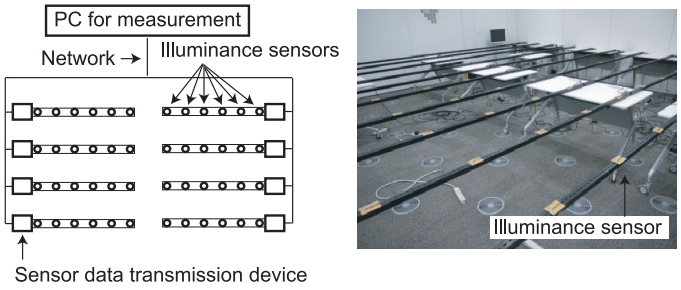


Fig. 3. Illuminance distribution measuring system

Sensor data transmission device in Fig.3-(a) has A/D converting function for illuminance value measured by illuminance sensors, and transmission function to transmit illuminance value to measurement PC. Therefore, it is possible to aggregate all measured illuminance data into the measurement PC by connecting sensor data transmission device to the measurement PC. Additionally, there is trade-off with measurement frequency of illuminance data through a hub, but in principle, there is no limit in the number of illuminance sensors that can be connected. Measurement PC visualizes illuminance distribution in real time based on the illuminance data received from the sensor data transmission device.

B. Illuminance Distribution Measurement Experiment

Illuminance distribution measurement experiment in actual office space was conducted by using illuminance distribution measurement system in order to evaluation of illuminance provided by the intelligent lighting system in actual office environment. A place for this experiment is "Ecozzeria" in Shin-Marunouchi Building mentioned in Chapter III-B. In this experiment, we use 204 illuminance sensors for measuring the illuminance distribution. Interval of illuminance sensors is set as 50 cm. The actual office has low partitions between desks and it is difficult to place illuminance distribution measurement system on desks. Therefore, illuminance sensors were installed at a position of 125 cm from the floor where there were no shielding objects. Illuminance at a height of 125 cm from the floor is discussed in the following. The experiment environment is shown in Fig.4, and experiment scene in Fig.5.

Two items were conducted in this study; an illuminance distribution measurement experiment under environment with uniform luminosity as lighting environment not using light control and illuminance distribution measurement under the intelligent lighting system environment. In the illuminance distribution measurement experiment under uniform luminosity environment, all lighting equipments' lighting luminosity was set to uniform luminosity which made illuminance of approximately 1,000 lx at a position of 125 cm from the floor. This illuminance is approximately 750 lx on the desk.

In illuminance distribution measurement experiment in Intelligent Lighting System environment, 10 illuminance sensors named A to J were installed at desks for control of Intelligent Lighting System. These illuminance sensors are respectively called from A to J. Target illuminance of each illuminance sensor is as shown in Fig.4.

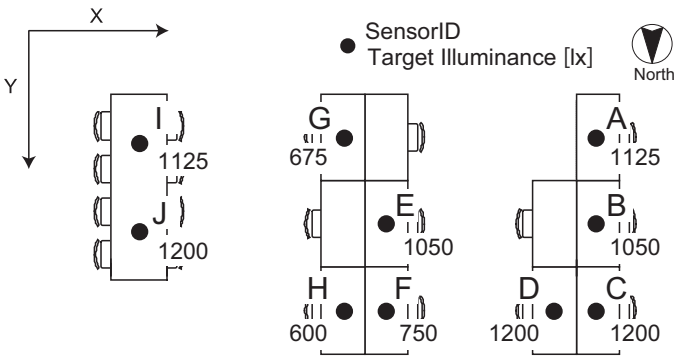


Fig. 4. Experiment environment

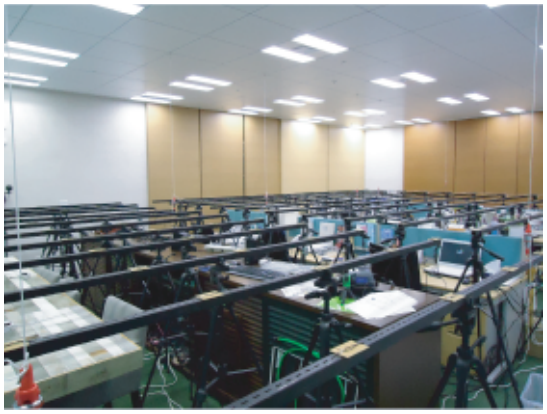


Fig. 5. Experiment scene

Fig.6 shows a result of illuminance distribution measurement experiment under uniform luminosity environment.

Furthermore, Fig.6 indicates that illuminance distribution under uniform luminosity environment is almost uniform illuminance environment, but areas directly under lighting equipments have higher illuminance. Areas beside the south windows have illuminance increased by 300 lx. It may be considered to be influenced by reflection from a wall, since lighting is installed close to a wall.

Next, the result of illuminance distribution measurement experiment in Intelligent Lighting System environment is shown Fig.7, and target illuminance and measured illuminance of each illuminance sensor are shown in Table.I, and average

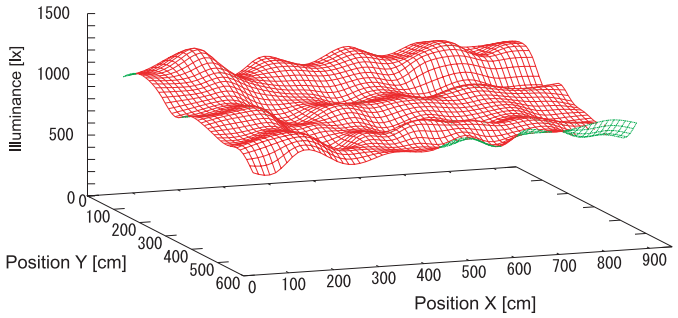


Fig. 6. Illuminance distribution (Uniform luminance)

illuminance at the desks with illuminance sensors installed and in other areas is shown in Table.II.

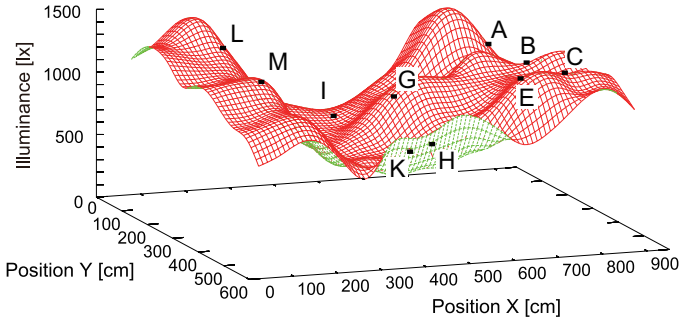


Fig. 7. Illuminance distribution (Using Intelligent Lighting System)

TABLE I
TARGET ILLUMINANCE AND MEASURED ILLUMINANCE OF EACH ILLUMINANCE SENSOR

Illuminance sensor	Target illuminance	Measured illuminance
A	1125	1169
B	1050	1195
C	1200	1249
D	1200	1240
E	1050	1004
F	750	781
G	675	695
H	600	708
I	1125	1174
J	1200	1193

TABLE II
AVERAGE ILLUMINANCE AT THE DESKS WITH ILLUMINANCE SENSORS INSTALLED AND IN OTHER AREAS

	Average illuminance
The desks with illuminance sensors installed	1174
Other Areas	936

Fig.7 indicates that the desks with illuminance sensors installed are provided with individual illuminance and the areas without illuminance sensors generally have lower illuminance in illuminance distribution in Intelligent Lighting System environment. In Table.I, difference between target illuminance and measured illuminance of almost all the illuminance sensors is within 50lx. Considering that illuminance difference which are acknowledgeable for humans in office environment is approximately 50 lx[12], target illuminance is assumed to be achieved for almost all the illuminance sensors. However, for Illuminance Sensors B and H, illuminance difference between target illuminance value and measured illuminance value was over 50 lx. This is because there are illuminance sensors of higher target illuminance values nearby. Illuminance Sensor B's target illuminance is 1050lx, but Illuminance Sensor A's target illuminance is 1125lx, and C's is 1200lx. Illuminance difference in Illuminance Sensor B occurred from demand to achieve these target illuminances. Furthermore, in terms of areas without illuminance sensors, Table.II shows that average illuminance of other areas is lower compared to average illuminance of desks with illuminance sensors installed. In Intelligent Lighting System, energy saving is achievable by setting illuminance lower in the areas where illuminance sensors are not installed.

Illuminance distribution in work areas is discussed next. Work areas mentioned here are desks provided to each worker. As illuminance distribution in work areas, illuminance distribution of Desk E that had especially greater illuminance change within the desk is shown in Fig.8.

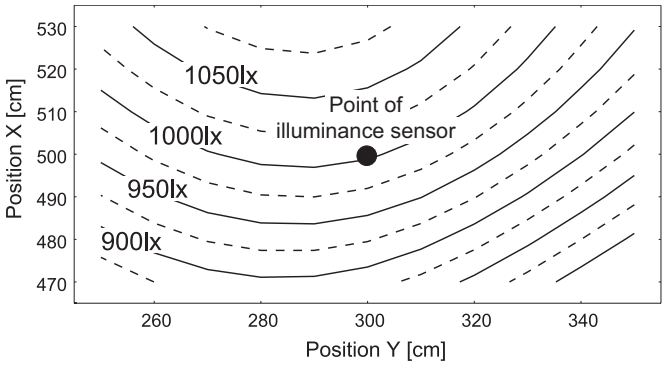


Fig. 8. Work area (Desk E)

Illuminance change within Desk E shown in Fig.8 was approximately 300 lx, and balance rate was 0.73. Balance rate of all other work areas was over 0.7. Considering that work area's balance rate recommended by CIE is over 0.7, illuminance environment of work areas provided by Intelligent Lighting System may be said as appropriate for work.

From the above results, in Intelligent Lighting System, the areas with illuminance sensors installed achieved required illuminance, and the areas without illuminance sensors had lower illuminance. In addition, illuminance change in work areas was indicated to be suitable environment for work in CIE standard.

IV. CONCLUSION

This study mentioned evaluation of illuminance provided by Intelligent Lighting System in actual office environment by using the system which can measure illuminance distribution. As a result, space provided by the intelligent lighting system

was appropriate for work, and areas not requiring illuminance provision had lower illuminance. Furthermore, illuminance distribution was formerly obtained by computer simulation in actual office space, but actual measurement helped obtain more precise illuminance distribution.

ACKNOWLEDGMENT

We are deeply grateful to Mr.Nishimoto, Mr.Omi of Mitsubishi Estate Co.,Ltd., Mr.Adachi of Mitsubishi Jisho Sekkei Inc., Mr.Matsumoto, Mr.Takei and Mr.Fukazawa of Sekonic Corporation and Dr.Ono of Ryukoku Univ.

REFERENCES

[1] M.Miki, and T.Kawaoka, Design of intelligent artifacts:a fundamental aspects, Proc. JSME International Symposium on Optimization and Innovative Design(OPID97), pp.1701-1707, September 1997
[2] Olli Seppanen, William J. Fisk: A Model to Estimate the Cost-Effectiveness of Improving Office Work through Indoor Environmental Control, Proceedings of ASHRAE, 2005
[3] M. J. Mendell, and G. A. Heath: Do indoor pollutants and thermal conditions in schools influence student performance? A critical review of the literature, Indoor Air, Vol.15, No.1, pp.27-52, 2005
[4] Peter R. Boyce, Neil H. Eklund, S. Noel Simpson, Individual Lighting Control: Task Performance, Mood, and Illuminance, Journal of the Illuminating Engineering Society, pp.131-142, Winter 2000
[5] M. Miki, T. Hiroyasu, and K.Imazato, Proposal for an intelligent lighting system, and verification of control method effectiveness, Proc. IEEE CIS, pp.520-525, 2004
[6] F. Kaku, M. Miki, T. Hiroyasu, M. Yoshimi, S. Tanaka, T. Nishida, N. Kida, M. Akita, J. Tanisawa, and T. Nishimoto, Construction of Intelligent Lighting System Providing Desired Illuminance Distributions in Actual Office Environment, Lecture Notes in Computer Science, Vol 6114, pp.451-460, 2010
[7] Miki M, Kasahara Y, Hiroyasu T, Yoshimi M, Construction of illuminance distribution measurement system and evaluation of illuminance convergence in intelligent lighting system, Proc IEEE SENSORS, pp.2431-2434, 2010
[8] CIE:Lighting of Indoor Work Places, CIE S 008/E-2001, p.4, 2001
[9] M.Miki,T.Hiroyasu,K.Imazato,Proposal for an intelligent lighting system,and verification of control method effectiveness, Proc IEEE CIS, pp.520-525, 2004
[10] S.Tanaka,M.Miki,T.Hiroyasu,M.Yoshikata,An Evolutional Optimization Algorithm to Provide Individual Illuminance in Workplaces, Proc IEEE Int Conf Syst Man Cybern, Vol.2, pp.941-947, 2009
[11] Mardaljevic, J., Daylight simulation: nalidation, sky models and daylight coefficients, PhD Thesis, De Montfort University, 2000
[12] T.Shikakura, H.Morikawa, and Y.Nakamura, Research on the Perception of Lighting Fluctuation in a Luminous Offices Environment, Journal of the Illuminating Engineering Institute of Japan, Vol.85 5, pp.346-351, 2001

Some Model Theoretical Results over Horn Formula

Maonian Wu[1], Mingyi Zhang[2], and Ying Zhang[2⋆]

[1] College of Science, Guizhou University, Guiyang, China.
[2] Guizhou Academy of Sciences, Guiyang, China.

Abstract. Horn clause plays a basic role in logic programming and are important for many constructive logics. Interpolation theorem is also an important topic for many monotonic or non-monotonic logic systems. The interpolation theorem for Horn clauses under the derivability from their proof theory was formulated by Gabby and Maksimova. Using model theoretical methods, this paper demonstrates the following results for Horn clauses under the classical propositional derivability. First we formulate that the restrict over a sub-language of a close set of Horn formulas is logically equivalent to a set of Horn formulas. Next the interpolation theorem and the parallel interpolation theorem over Horn formulas under classical propositional derivability was formulated. The paper answers partially the open problem proposed by Kourousias and Makinson in 2007.
Key words: Horn Formula, Interpolation Theorem, Model Theory.

1 Introduction

A Horn clause, a clause (a disjunction of literals) with at most one positive literal, is first pointed out by the logician Alfred Horn from its significance [Hor51]. In 1965, Keisler proposed that Horn theories are characterized semantically by the fact that the models of a Horn theory are closed under intersections of positive atoms in an interpretation[CK90]. Khardon formulated the characteristic models of a Horn theory [Kha95].

Horn clause play a basic role in logic programming and are important for constructive logic. Horn logic is equivalent in computational power to a universal Turing machine. In fact, the resolution of a goal clause with a definite clause to produce a new goal clause is the basis of the SLD resolution inference rule, used to implement logic programming and the programming language Prolog. In logic programming a definite clause behaves as a goal-reduction procedure.

It is well-known that computational intractability is one of the major challenges for many topics in computer science. For example there is computational intractability in belief revision [AGM85]. most of belief change operations are computationally intractable if knowledge and beliefs are represented in propositional languages [Neb92,EG92]. Although tractability is not taken for granted when the language is restricted to Horn formulae. Currently more and more researchers considered belief revision upon Horn logic [Del08,BMVW10,DW10] since its computation is better than proposition logic. Only a example in belief revision, we can know that Horn logic is more and more important for many researches based on logic.

⋆ Corresponding Email: Zhangying@yahoo.com.cn(Ying Zhang).

In mathematical logic, Craig's interpolation theorem is a important result about the relationship between different logical theories [GM05]. Roughly stated, the theorem says that if a formula φ implies a formula ψ then there is a third formula θ, called an interpolant, such that every nonlogical symbol in θ occurs both in φ and ψ, φ implies θ, and θ implies ψ. The theorem was first proved for first-order logic by Craig in 1957. Variants of the theorem hold for other logics, such as propositional logic. A stronger form of Craig's theorem for first-order logic was proved by Lyndon in 1959; the overall result is sometimes called the Craig-Lyndon theorem.

Parallel interpolation theorem upon classical propositional logic was formulated in [KM07]. It also is a very important for splitting for proposition logic and other logic systems [KM07,WZZ08,WZ10,WZ08]. An open question was posted in [KM07]: *How far can the results be established for sub-classical (e.g. intuitionistic) consequence relations or supra-classical ones?* So (parallel) interpolation theorem upon Horn logic is important. Using model theoretical methods, this paper demonstrates the following results for Horn clauses. First the restrict of a close set of Horn formulas over a sub-language is classical logically equivalent to some set of Horn formulas was formulated. Next the interpolation theorem and the parallel interpolation theorem over Horn formulas under classical propositional derivability was formulated.

The rest of the paper is organized as follows: In section 2, we recall many preliminaries for the paper. We show that model theoretical method about interpolation theorem over set of Horn formulas in section 3. In the next section, we compare our results with ones of some related works. The conclusion of the paper will discuss in section 5.

2 Preliminaries

We assume a propositional language $\mathcal{L}$ with a countable set Φ of elementary letters (alias propositional variables), the standard logical connectives $\neg, \vee, \wedge, \rightarrow$, and the logical constants $\top$ (**true**) and $\bot$ (**false**). We use lower case letters $a, b, ..., x, y, z, \alpha, \beta, ...$ to range over formulae of classical propositional logic. Sets of formulae are denoted by upper case letters $A, B, ..., X, Y, ...$, reserving L for the set of all formulae, E for the set of all elementary letters (alias propositional variables) and $F, E', E_1, E_2, \ldots$ for subsets of the elementary letters. For any formula α, we write $E(\alpha)$ to mean the set of the elementary letters occurring in α; similarly for sets A of formulae. Let $F \subseteq E$, $L(F)$ stands for the sub-language generated by F, i.e. the set of all formulae x with $E(x) \subseteq F$. Classical consequence is written as $\vdash$ when treated as a relation over $2^L \times L$, classical consequence operation is written as Cn when treated as an operation on 2^L into itself. The relation of classical equivalence is written $\dashv\vdash$. We say that set K of formulae is belief set if it is close, i.e., $K = Cn(K)$.

Let $v : \Phi \rightarrow \{0, 1\}$ be a valuation of $\mathcal{L}$. To lighten notations, $v(\varphi)$ represents also the truth value of formula φ under the standard interpretation of the propositional connectives. Moreover, we say that $M = \{p \in \Phi : v(p) = 1\}$ is a model of a formula φ (or set of formulae), written by $M \models \varphi$, whenever valuation v of $\mathcal{L}$ makes φ true. Evidently, a model M of a formula is a subset of set Φ of elementary letters , i.e., $M \subseteq \Phi$. A formula (or set of formulae) is consistent iff it has a model. for any set F

of formulae, $v(F) = 1$ means that $v(\varphi) = 1$ for all $\varphi \in F$ while $v(F) = 0$ means $v(\varphi) = 0$ for some $\varphi \in F$. We write $Mod(\varphi)$ to denote the set of valuations that make formula φ (or set of formulae) to be true.

We will recall the well-known Craig's interpolation theorem about propositional logic.

Theorem 1 *(Craig's Interpolation Theorem over propositional logic [Cra57]) Let $A \subseteq L$ and formula ψ. If $A \vdash \psi$ then there is $\phi \in L$ such that $A \vdash \phi$, $\phi \vdash \psi$ and every sentence symbol which occurs in ϕ also occurs in both A and ψ.*

When A is a set of Horn formulas and ψ is a Horn formula, interpolation formula ϕ in Craig's interpolation theorem logically equivalent to a CNF formula. In other words, ϕ maybe is a CNF formula. But a CNF formula is often not a Horn formula.

3 Interpolation Theorem in Horn formulas

Since the perfect computation of Horn formula, it is very important in the research of the proposition logic and logic program and so on. In the section, the restrict of a close set of Horn formulas over a sub-language is logically equivalent to some set of Horn formulas was formulated. Next we show the interpolation theorem over Horn formulas. Finally the parallel interpolation theorem over Horn formulas was formulated. The paper answers a part of the open problem proposed in [KM07].

First we will recall the well-known relation of model's intersection and a set of Horn formulas.

Theorem 2 *(Characterization of Horn Formula [CK90])*

- *Set A of formulas is logically equivalent to some set of Horn clauses if and only if $M_1, M_2 \in Mod(A)$ then $M_1 \cap M_2 \in Mod(A)$.*
- *Formula φ is logically equivalent to some set of Horn clauses if and only if $M_1, M_2 \in Mod(\varphi)$ then $M_1 \cap M_2 \in Mod(\varphi)$.*

Now we show that the consequence of a set of Horn Formula under a sub-language is logically equivalent to some set of Horn Formula over the sub-language by the methods of model theory.

Theorem 3 *Let $A \subseteq Horn(L)$ and $P^{'}$ be any subset of the set of elementary letters of the proposition logic. Then $\{\phi | A \vdash \phi$ and $E(\phi) \subseteq P^{'}\}$ is logically equivalent to some set of Horn Formulas.*

Proof: Let $\sum = \{\phi | A \vdash \phi$ and $E(\phi) \subseteq P^{'}\}$. Suppose m_1, m_2 are any two models of $\sum$ over $P^{'}$. Let $Th(m_1) = \{\varphi | m_1 \models \varphi \quad \& \quad E(\varphi) \subseteq P^{'}\}$ and $Th(m_2) = \{\varphi | m_2 \models \varphi \quad \& \quad E(\varphi) \subseteq P^{'}\}$. Then $A \cup Th(m_1)$ is consistent. (Otherwise, there are $\varphi_1, ..., \varphi_n \in Th(m_1)$ such that $A, \varphi_1, ..., \varphi_n \vdash \bot$, i.e. $A \vdash \neg(\varphi_1 \wedge ... \wedge \varphi_n)$. Hence $\neg(\varphi_1 \wedge ... \wedge \varphi_n) \in \sum \subseteq Th(m_1)$. So it is a contradiction since $\neg(\varphi_1 \wedge ... \wedge \varphi_n) \in Th(m_1)$ and $\varphi_1, ..., \varphi_n \in Th(m_1)$). Similarly, $A \cup Th(m_2)$ is consistent. Then there are models M_1 and M_2 over the entire set of elementary letters

of the proposition logic such that $M_1 \in Mod(A \cup Th(m_1))$ and $M_2 \in Mod(A \cup Th(m_2))$. Clearly, $M_1, M_2 \in Mod(A)$. By Theorem 3.1, we have $M_1 \cap M_2 \in Mod(A)$ since $A \subseteq Horn(L)$. Since $M_1, M_2 \in Mod(A)$ and the construction of $\sum$ we have $M_1, M_2, M_1 \cap M_2 \in Mod(\sum)$. So we have $M_1 \restriction_{P'} \in Mod(\sum)$, $M_2 \restriction_{P'} \in Mod(\sum)$ and $M_1 \cap M_2 \restriction_{P'} \in Mod(\sum)$ since $E(\sum) \subseteq P^{'}$. It is clear that $M_1 \restriction_{P'} = m_1$ and $M_2 \restriction_{P'} = m_2$ since $M_1 \in Mod(A \cup Th(m_1))$ and $M_2 \in Mod(A \cup Th(m_2))$. Hence $m_1 \cap m_2 = (M_1 \restriction_{P'}) \cap (M_2 \restriction_{P'}) = (M_1 \cap M_2) \restriction_{P'} \in mod(\sum)$. By Theorem 3.1, we show that $\sum = \{\phi | A \vdash \phi \text{ and } E(\phi) \subseteq P^{'}\}$ is logically equivalent to some set of Horn formulas. □

The Theorem is a important base of interpolation theorem in Horn formula.

When A is a set of Horn formulas and ψ is a Horn formula, interpolation formula ϕ in Craig's interpolation theorem logically equivalent to a CNF formula. In other words, ϕ maybe is a CNF formula. But a CNF formula is often not a Horn formula. So we will propose interpolation theorem over Horn formulas under classical propositional derivability as Craig's.

Theorem 4 *(Interpolation Theorem over Horn formulas) Let $A \subseteq Horn(L)$ and $\psi \in L$. If $A \vdash \psi$ then there is a Horn formula ϕ such that $A \vdash \phi \vdash \psi$ and every sentence symbol which occurs in ϕ also occurs in both A and ψ.*

Proof: By Theorem 3.2, we have $S = \{\phi | A \vdash \phi \text{ and } E(\phi) \subseteq (E(A) \cap E(\psi))\}$ is logically equivalent to some set $\sum$ of Horn formulas over $(E(A) \cap E(\psi))$ since $A \subseteq Horn(L)$. Clearly, $A \vdash \sum$. Since $A \vdash \psi$, there exists $\phi \in L$ such that $A \vdash \phi, \phi \vdash \psi$ and every sentence symbol which occurs in ϕ also occurs in both A and ψ by interpolation theorem. So $\phi \in S$. Clearly, $\sum \vdash \phi$ since S is logically equivalent to $\sum$. Then $\sum \vdash \psi$. Hence there are finite Horn clause $\phi_1, ..., \phi_n \in \sum$ such that $\phi_1, ..., \phi_n \vdash \psi$ and $E(\phi_i) \subseteq (E(A) \cap E(\psi))$ for all $i = 1, ..., n$. It is easy that the theorem is correct when let $\phi = \phi_1 \wedge ... \wedge \phi_n$. □

Note that consequence ψ may be non-Horn formula. The theorem is evidently correct when ψ is a Horn formula.

Corollary 1 *Let $A \subseteq Horn(L)$ and $\psi \in Horn(L)$. If $A \vdash \psi$ then there is a Horn formula ϕ such that $A \vdash \phi \vdash \psi$ and every sentence symbol which occurs in ϕ also occurs in both A and ψ.*

Although Parallel Interpolation Theorem in Horn Formula by the deduction from their proof theory [WZZ11]. Now we will show the important result, Parallel Interpolation Theorem in Horn Formula under classical propositional derivability. It is important base of the finest splitting of any set of horn formulas under classical propositional derivability in future.

Theorem 5 *(Parallel Interpolation Theorem in Horn Formula) Let $A = \bigcup\{A_i\}_{i \in I} \subseteq Horn(L)$ where the letter sets $E(A_i)$ are pairwise disjoint, and $\bigcup\{A_i\}_{i \in I} \vdash \psi$. Then there are Horn formula ϕ_i such that $A_i \vdash \phi_i$ for any $i \in I$. $\bigcup_{i \in I}\{\phi_i\} \vdash \psi$ and every sentence symbol which occurs in ϕ_i also occurs in both A_i and ψ for all i.*

Proof: The proof is similar to [WZZ11]. We omit it since space's limitation of the paper. □

4 Compare with related works

First we give the structure of Horn formula and a proof theory for Horn formula. Following [DW10], Horn formulae are defined as follows:

1. Every $a \in \Phi$ is a Horn clause.
2. $a_1 \wedge a_2 \wedge ... \wedge a_n \rightarrow a$ is a Horn clause, where $n \geq 0$ and $a, a_i \in \Phi$ $(1 \leq i \leq n)$.
3. Every Horn clause is a Horn formula.
4. If φ and ψ are Horn formulae, so is $\varphi \wedge \psi$.

We let $Horn(\mathcal{L})$ be the set of all Horn formulae with respect to $\mathcal{L}$.

A Horn formula ψ is derived from a set F of Horn formulae, written as $F \vdash_H \psi$, if ψ can be obtained from F by a finite number of applications of the following rules and axioms:

Axioms:

$\bot \rightarrow a \qquad a \rightarrow a$

Rules:

1. From $a_1 \wedge a_2 \wedge ... \wedge a_n \rightarrow a$ and $b_1 \wedge ... \wedge b_m \rightarrow a_i$ infer $a_1 \wedge a_2 \wedge ... \wedge a_{i-1} \wedge b_1 \wedge b_2 \wedge ... \wedge b_m \wedge a_{i+1} \wedge ... \wedge a_n \rightarrow a$.
2. From $a_1 \wedge a_2 \wedge ... \wedge a_n \rightarrow a$ infer $a_1 \wedge a_2 \wedge ... \wedge a_n \wedge b \rightarrow a$.
3. For rules $r_1 := a_1 \wedge a_2 \wedge ... \wedge a_n \rightarrow c$ and $r_2 := b_1 \wedge b_2 \wedge ... \wedge b_m \rightarrow d$, if $\{a_1, a_2, ..., a_n\} = \{b_1, b_2, ..., b_m\}$ and $c = d$ then from r_1 infer r_2.
4. From $\phi \wedge \psi$ infer ϕ and ψ.
5. From ϕ, ψ infer $\phi \wedge \psi$.

where a, b, c (with or without subscriptions) range over the elementary letters only. For two sets P and Q of Horn formulae, $P \vdash_H Q$ means $P \vdash_H \varphi$ for all $\varphi \in Q$.

A set F of Horn formulae is inconsistent if $F \vdash_H \bot$ and is consistent otherwise. If the set $H(F) = \{\varphi \in Horn(\mathcal{L}) : F \vdash_H \varphi\}$ then we call $H(F)$ is *the Horn closure of* F.

Gabby and Maksimova formulated the interpolation theorem for Horn clauses by the deduction from their proof theory. The proof see Theorem 15.8 in [GM05]. We can verify that it remains true under the current proof theory.

Theorem 6 *(Interpolation theorem for Horn logic [GM05]) Let P and Q be sets of Horn clauses and a be an elementary letter in $E(Q)$. Suppose that $P, Q \vdash_H a$. Then there is a set H of Horn clauses in $L(P) \cap L(Q)$, such that $P \vdash_H H$ and $H, Q \vdash_H a$.*

Proof: By inductive on the complexity of the proof.

Base step:

Case 1: $r \in P \cap Q$

The result holds whenever let $H = \{r\}$.

Case 2: $r \notin Q$ and $r \in P$

We have $r \in L_1 \cap L_2$ since r be an elementary letter in L_2 and $r \in P \subseteq L_1$ The result holds whenever let $H = \{r\}$.

Case 3: $r \in Q$ and $r \notin P$

The result holds whenever let $H = \{$elementary letter $l \in L_1 \cap L_2 : P \vdash_H l\}$ or $H = \{\top\}$.

Induction step:

Case 4: $r \notin Q$ and $r \notin P$

There is a Horn clause $a_1 \wedge a_2 \wedge ... \wedge a_n \rightarrow r \in P \cup Q$ such that $P \cup Q \vdash_H \{a_1, a_2, ..., a_n\}$ since $P, Q \vdash_H r$[3]. That is $P \cup Q \vdash_H a_i$ for any $i \in \{1, 2, .., n\}$.

If $a_1 \wedge a_2 \wedge ... \wedge a_n \rightarrow r \in Q$ then a_i is an elementary letter in L_2 for any i. By inductive, there are set H_i of Horn clauses such that $E(H_i) \subseteq L_1 \cap L_2$, $P \vdash_H H_i$ and $H_i, Q \vdash_H a_i$ for each i. Let $H = \bigcup_i H_i$. Then $E(H) \subseteq L_1 \cap L_2$, $P \vdash_H H$ and $H, Q \vdash_H a_1 \wedge a_2 \wedge ... \wedge a_n$. So the results holds.

If $a_1 \wedge a_2 \wedge ... \wedge a_n \rightarrow r \notin Q$ then $a_1 \wedge a_2 \wedge ... \wedge a_n \rightarrow r \in P$. The result also holds when $a_1, .., a_i, ..., a_n$ are elementary letters in L_2 by the above discussion and added Horn formula $a_1 \wedge a_2 \wedge ... \wedge a_n \rightarrow r$ in H. Suppose that only $a_i \notin L_2$ for some i. If $a_i \in P$ then $a_1 \wedge a_2 \wedge ... \wedge a_{i-1} \wedge a_{i+1} \wedge ... \wedge a_n \rightarrow r \in P$ since $a_1 \wedge ... \wedge a_i \wedge ... \wedge a_n \rightarrow r \in P$ and $a_i \in P$. The result holds since $a_1, a_2, ..., a_{i-1}, a_{i+1}, ..., a_n$ are elementary letters in L_2. If $a_i \notin P$ then there is a Horn clause $b_1 \wedge b_2 \wedge ... \wedge b_m \rightarrow a_i \in P \cup Q$ such that $P \cup Q \vdash_H b_1, b_2, ..., b_m$ since $P \vdash_H a_i$. The result holds for finite iterate the procedure as r because Horn clause is finite construction and the procedure end in finite steps. Suppose there are $a_{i_1}, .., a_{i_l} \notin L_1$. the result holds by iterate the prevenient procedure for each i_j. □

Corollary 2 *Let $A \subseteq Horn(L)$ and $\psi \in Horn(L)$. If $A \vdash_H \psi$ then there is a Horn formula ϕ such that $A \vdash_H \phi \vdash_H \psi$ and every sentence symbol which occurs in ϕ also occurs in both A and ψ.*

Evidently, Rules of Horn logic in section 2 is correct in classical proposition logic. So Corollary 2 hold when we drop the subscript H for the Horn derivability symbol $\vdash_H$.

From the section, we know that Gabbay and Maksimova's interpolation theorem is better if consequence ψ in interpolation theorem is a Horn formula. But Gabbay and Maksimova's interpolation theorem work out when consequence ψ is not a Horn formula.

5 Conclusion And Future Works

Using model theoretical methods, this paper demonstrates the following three results for Horn clauses. The restrict of a close set of Horn formulas over a sub-language is classical logically equivalent to some set of Horn formulas was formulated. And we show the Craig and parallel interpolation theorem over Horn formulas under classical

[3] The case, there are elementary letters $\{a_1, a_2, ..., a_n\} \subseteq P \cup Q$ such that $P \cup Q \vdash_H a_1 \wedge a_2 \wedge ... \wedge a_n \rightarrow r$, may exist. Since $P \cup Q \vdash_H a_1 \wedge a_2 \wedge ... \wedge a_n \rightarrow r$ obtained by only using the Rule 1,2,3, we know that any rule in Rule 1,2,3 need a Horn formula $b_1 \wedge b_2 \wedge ... \wedge b_m \rightarrow r \in P \cup Q$. So we only discuss the above case

propositional derivability. We answer partially the open problem proposed by Kourousias and Makinson in 2007.

In future, we will research the finest splitting and local belief revision based on this parallel interpolation theorem.

6 Acknowledgements

We are grateful to Dr. Dongmo Zhang and two unknown referees for their insightful suggestions and objective argumentation. This work is Supported partially by the National Science Foundation of China under Grant No.61003203, Natural Science Foundation of Guizhou province No.[2009]2123, Science research fund for talents recruiting of Guizhou university No.(2008)033 and Natural Science Youth Foundation of Guizhou University(No.2009019).

References

[AGM85] Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson. On the logic of theory change: Partial meet contraction and revision functions. *J. Symb. Log.*, 50(2):510–530, 1985.

[BMVW10] Richard Booth, Thomas Meyer, Ivan José Varzinczak, and Renata Wassermann. Horn belief change: A contraction core. In *ECAI-10*, pages 1065–1066, 2010.

[CK90] C.C. Chang and H.J. Keisler. *Model Theory*. North Holland, 1990.

[Cra57] William Craig. Three uses of the herbrand-gentzen theorem in relating model theory and proof theory. *The Journal of Symbolic Logic*, 3(22):269–285, 1957.

[Del08] James P. Delgrande. Horn clause belief change: Contraction functions. In *KR-08*, pages 156–165, 2008.

[DW10] James P. Delgrande and Renata Wassermann. Horn clause contraction functions: Belief set and belief base approaches. In *KR-10*, pages 143–152, 2010.

[EG92] Thomas Eiter and Georg Gottlob. On the complexity of propositional knowledge base revision, updates, and counterfactuals. *Artificial Intelligence*, 57(2-3):227–270, October 1992.

[GM05] Dov M Gabbay and Larisa Maksimova. *Interpolation and definability: modal and intuitionistic logics*. Clarendon Press, 2005.

[Hor51] Alfred Horn. On sentences which are true of direct unions of algebras. *The Journal of Symbolic Logic*, 1(16):14–21, 1951.

[Kha95] Roni Khardon. Translating between horn representations and their characteristic models. *Journal of Artificial Intelligence Research*, (3):349–372, 1995.

[KM07] George Kourousias and David Makinson. Parallel interpolation, splitting, and relevance in belief change. *J. Symb. Log.*, 72(3):994–1002, 2007.

[Neb92] B. Nebel. Syntax-based approaches to belief revision. In Gärdenfors, editor, *Belief Revision*, pages 52–88. Cambridge University Press, 1992.

[WZ08] Maonian Wu and Mingyi Zhang. Axiom p and reliable belief change operator in belief revision. In *IC-AI*, pages 165–171, 2008.

[WZ10] Maonian Wu and Mingyi Zhang. Algorithms and application in decision-making for the finest splitting of a set of formulae. *Knowl.-Based Syst.*, 23(1):70–76, 2010.

[WZZ08] Maonian Wu, Zhaohui Zhu, and Mingyi Zhang. Maximal invariable update operator based on splitting. In *IMECS2008*, pages 7–12, 2008.

[WZZ11] Maonian Wu, Dongmo Zhang, and Mingyi Zhang. Language splitting and relevance-based belief change in horn logic. In *Proceedings AAAI-11*, page to appear, 2011.

An Automated Derivation of Łukasiewicz's *CN* from the Hilbert/Ackerman *Grundzüge* Sentential Calculus

Jack K. Horner
P. O. Box 266
Los Alamos, New Mexico 87544 USA

Abstract

Two logics are implicationally equivalent if the axioms and inference rules of each imply the axioms of the other. Characterizing the inferential equivalences of various formulations of the sentential calculi is thus foundational to the study of logic. Using an automated deduction system, I show that Łukasiewicz's CN can be derived from the sentential calculus, here called GTL, of Hilbert/Ackerman's Grundzüge der theoretischen Logik. *Although each of these systems is known to imply the other, the proof presented here appears to be novel.*

Keywords: propositional logic, automated deduction, sentential calculus

1.0 Introduction

Two logics are implicationally equivalent if the axioms and inference rules of each imply the axioms of the other. Characterizing the equivalences of various formulations of the sentential calculi is thus foundational to the study of logic ([1],[3]-[7],[9]-[10],[12]-[15]).

"CN", the formulation of the sentential calculus in [1], is among the most compact: its vocabulary contains only two logical connectives (C, and N) and sentence variables (*p, q, r,* ...). It has two inference rules (condensed detachment and substitution), and three axioms.

In CN, any expression of the form *Cxy* or *Nz*, where *x, y,* and *z* are sentences, is a sentence. *Cpq* is interpreted as "sentence *p* implies sentence *q*"; *Np* is interpreted as "not-*p*". N has higher associative precedence than C. For example,

CCqrCpNr

translates to the more common "arrow-and-parenthesis" notation as

$$(q \rightarrow r) \rightarrow (p \rightarrow \sim r)$$

where "$\rightarrow$" designates "implies" and "~" designates "not".

The axioms of CN in [1] are:

CN1. *CCpqCCqrCpr*
CN2. *CCNppp*
CN3. *CpCNpq*

Cast in CN notation, the axioms of the Hilbert/Ackerman *Grundzüge* sentential calculus (GTL, [9]) are (corresponding CN theorems are identified in parentheses):

H1. *CxCyx* (CN18)
H2. *CCxCxyCxy* (CN30)
H3. *CCxCyzCyCxz* (CN21)
H4. *CCyzCCxyCxz* (CN22)
H5. *CpCNpq* (CN3)
H6. *CCpqCCNpqq* (CN54)

The main result of this paper is that the axioms of [9] collectively imply the axioms of [1].

2.0 Method

To show that the axioms of [9] imply the axioms of [1], the *prover9* ([2]) script shown in Figure 1 was executed under on a Dell Inspiron 545 with an Intel Core2 Quad CPU Q8200 @ 2.33 GHz and 8.00 GB RAM, running under the *Windows Vista Home Premium (SP2)/Cygwin* operating environment.

```
set(hyper_resolution).

formulas(usable).
P ( i(x, i(y,x)) )                              # label("AxH1").
P ( i(i(x, i(x,y)), i(x,y))  )                  # label("AxH2").
P ( i(i(x, i(y,z)), i(y, i(x,z))) )             # label("AxH3").
P ( i(i(y,z), i(i(x,y), i(x,z)))  )             # label("AxH4").
P ( i(x, i(-x,y)) )                             # label("AxH5").
P ( i(i(x,y), i(i(-x,y),y)) )                   # label("AxH6").
end_of_list.

formulas(sos).
-P(i(x,y)) | -P(x) | P(y)          # label("InfConDet").
end_of_list.

formulas(goals).
P( i(i(x,y), i(i(y,z), i(x,z))) )       # label("AxCN1").
P( i(i(-x,x), x) )                      # label("AxCN2").
P( i(x, i(-x,y)) )                      # label("AxCN3").
end_of_list.
```

Figure 1. The *prover9* script (in Horn clause ([11]) form), used to show that the axioms of GTL imply the axioms of CN. The inference rules are condensed detachment and substitution. The implementation of condensed detachment is the formula in the "sos" list; substitution is derived from *prover9*'s hyperresolution rule (introduced in the "set" command at the top of the script). Details of *prover9*'s syntax and semantics can be found in [2].

3.0 Results

Figure 3 shows that GTL implies CN.

```
============================== PROOF =================================

% Proof 1 at 0.01 (+ 0.03) seconds: "AxCN3".
% Length of proof is 4.
```

```
% Level of proof is 2.
% Maximum clause weight is 7.
% Given clauses 0.

3 P(i(x,i(-x,y))) # label("AxCN3") # label(non_clause) # label(goal).
[goal].
8 P(i(x,i(-x,y))) # label("AxH5").  [assumption].
13 -P(i(c5,i(-c5,c6))) # label("AxCN3") # answer("AxCN3").  [deny(3)].
14 $F # answer("AxCN3").  [resolve(13,a,8,a)].

============================== end of proof ==========================

============================== PROOF =================================

% Proof 2 at 0.01 (+ 0.03) seconds: "AxCN1".
% Length of proof is 7.
% Level of proof is 2.
% Maximum clause weight is 12.
% Given clauses 1.

1 P(i(i(x,y),i(i(y,z),i(x,z)))) # label("AxCN1") # label(non_clause) #
label(goal).  [goal].
6 P(i(i(x,i(y,z)),i(y,i(x,z)))) # label("AxH3").  [assumption].
7 P(i(i(x,y),i(i(z,x),i(z,y)))) # label("AxH4").  [assumption].
10 -P(i(x,y)) | -P(x) | P(y) # label("InfConDet").  [assumption].
11 -P(i(i(c1,c2),i(i(c2,c3),i(c1,c3)))) # label("AxCN1") #
answer("AxCN1").  [deny(1)].
34 P(i(i(x,y),i(i(y,z),i(x,z)))).  [hyper(10,a,6,a,b,7,a)].
35 $F # answer("AxCN1").  [resolve(34,a,11,a)].

============================== end of proof ==========================

============================== PROOF =================================

% Proof 3 at 0.01 (+ 0.03) seconds: "AxCN2".
% Length of proof is 9.
% Level of proof is 3.
% Maximum clause weight is 8.
% Given clauses 4.

2 P(i(i(-x,x),x)) # label("AxCN2") # label(non_clause) # label(goal).
[goal].
4 P(i(x,i(y,x))) # label("AxH1").  [assumption].
5 P(i(i(x,i(x,y)),i(x,y))) # label("AxH2").  [assumption].
9 P(i(i(x,y),i(i(-x,y),y))) # label("AxH6").  [assumption].
10 -P(i(x,y)) | -P(x) | P(y) # label("InfConDet").  [assumption].
12 -P(i(i(-c4,c4),c4)) # label("AxCN2") # answer("AxCN2").  [deny(2)].
39 P(i(x,x)).  [hyper(10,a,5,a,b,4,a)].
51 P(i(i(-x,x),x)).  [hyper(10,a,9,a,b,39,a)].
52 $F # answer("AxCN2").  [resolve(51,a,12,a)].

============================== end of proof ==========================
```

Figure 2. Summary of a *prover9* ([2]) proof showing that GTL ([9]) implies CN ([1]).

The total time to complete the proofs shown in Figure 2 was ~ 0.1 seconds on the platform described in Section 2.0.

4.0 Conclusions and discussion

Section 3 demonstrates that GTL implies CN. A companion paper ([16]) proves CN implies GTL. With the exception of the proof of CN3, the proof in Figure 2 appears to be novel.

The proof of CN3 is trivial because it is identical to H5. H3 and H4 are sufficient to derive CN1; H1, H2, and H6 are sufficient to derive CN2.

5.0 References

[1] Łukasiewicz J. *Elements of Mathematical Logic*. Second Edition (1958). Trans. by Wojtasiewicz O. Pergamon Press. 1963.

[2] McCune WW. *prover9 and mace4.* http://www.cs.unm.edu/~mccune/prover9/. 2009.

[3] Aristotle. *Prior Analytics*. Trans. by A. J. Jenkinson. In Aristotle. *The Basic Works of Aristotle.* Ed. by R. McKeon. Random House. 1941. pp. 62-107.

[4] Aristotle. *Posterior Analytics.* Trans. by G. R. G. Mure. In Aristotle. *The Basic Works of Aristotle.* Ed. by R. McKeon. Random House. 1941. pp. 108-186.

[5] Tarski A. *Introduction to Logic*. Trans. by O. Helmer. Dover. 1941.

[6] Hempel C. Studies in the logic of explanation. In Hempel C. *Aspects of Scientific Explanation and Other Essays in the Philosophy of Science.* Free Press. 1965. pp. 245-290.

[7] Quine WVO. *Philosophy of Logic*. Second Edition. Harvard. 1986.

[8] Chang CC and Keisler HJ. *Model Theory*. North-Holland. 1990.

[9] Hilbert D and Ackerman W. *Grundzüge der theroretischen Logik.* Second Edition. Dover. 1946.

[10] Frege G. Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens. Halle. 1879. Translated in van Heijenoort J. Begriffsschrift, a formula language, modeled upon that of arithmetic, for pure thought. *From Frege to Gödel: A Source Book in Mathematical Logic, 1879-1931*. Harvard. 1967. pp. 3-82.

[11] Horn A. On sentences which are true of direct unions of algebras. *Journal of Symbolic Logic* 16 (1951), 14–21.

[12] Church A. *Introduction to Mathematical Logic.* Volume I. Princeton. 1956.

[13] Birkhoff G and von Neumann J. The logic of quantum mechanics. *Annals of Mathematics* 37 (1936), 823-843.

[14] Kant I. *Kant's Introduction to Logic* (1800). Trans. by Abbott TK. Greenwood Press. 1963.

[15] Cohen MR and Nagel E. *An Introduction to Logic and Scientific Method.* Harcourt, Brace, and Company. 1934.

[16] Horner JK. An automated derivation of the Hilbert/Ackerman *Grundzüge der theoretischen Logik* sentential calculus from Łukasiewicz's *CN*. *Proceedings of the 2011 International Conference on Artificial Intelligence.* CSREA Press. Forthcoming.

MAKER: A New Algorithm in Finding Frequent Itemsets

Dr. Masoud Yaghini[1], **Kaveh Rasouli Chizari**[2], **Mahsa Mortazavi**[2], **Erfan Khaji**[3], **and Mahyar Hoseynzadeh**[3]

[1] School of Railway Eng., Iran University of Science & Technology, Tehran, Iran
[2] School of IT & Electronics, Iran University of Science & Technology, Tehran, Iran
[3] School of Railway Eng., Iran University of Science & Technology, Tehran, Iran

Abstract *- Many frequent itemset algorithms have been proposed within recent years most of which have some difficulties when minimum support is low or the dataset is dense. In this paper, we present a novel algorithm named Maker using data storing based on prime numbers and Dynamic Blocking which addresses the blocks in which a product is purchased with at least α probability. We show that dynamic blocking drastically cut down the size of memory required to store dense datasets. We demonstrate how maker incorporated into previous mining methods increase the performance significantly.*[1]

Keywords: Data mining, Association rules, Dynamic-Blocking, Maker algorithm

1 Introduction

Frequent Itemsets Mining (FIM) is a classical issue in data mining which has been the subject of many investigations within recent years. However, there is still one unsolved problem which is FIM in dense datasets with low minimum support. Historically, investigating on three major areas has led researchers to novel approaches to optimize former algorithms: Code optimization, candidate generation and data structure. Most reputed Apriori [1] has presented a practical candidate generation approach used by many later procedures, Patricia, kdci and lcm [2, 3, 4] to name a few. The idea of changing data structure was first discussed in Fp-growth [5] and developed in Eclat [6] and their descendants [7], [9]. Beside vertical and horizontal datasets, Prime Number dataset [10] is also considered as useful mean to decrease the time consumed in referring the main dataset. As Apriori and the other algorithms which investigated candidate generation need refer to the main memory for many times, they usually demands the other approaches such as data structure to achieve a satisfactory performance. Moreover, decreasing the value of minimum support will result in enlarging the volume of all of the data structures which would make trouble in generating and searching both the structure and candidates. Hence, the so-called problem cannot be handled effectively by the previous algorithms. In this paper, we present an innovative approach named Maker using two novel procedures to solve the problem, *Dynamic Blocking* and *Prime number* dataset which led us to a perfect result in mining dense datasets with low minimum support.

1.1 Problem Statement

Let us fix notations for the frequent itemset mining problem in the rest of this section. Let A be a set, called *set of items* or *alphabet*. Any subset $X \in \rho(A)$ of A is called an *itemset*. Let $\Gamma \subseteq \rho(A)$ be a multiset of itemsets, called *transaction database*, and its elements $T \in \Gamma$ called *transactions*. For a given itemset $X \in \rho(A)$, the set of transactions that contain X

$$\Gamma(X) := \{ T \in \Gamma | X \subseteq T \} \quad (1)$$

is called *(transaction) cover of X in* Γ and its cardinality

$$\mathrm{Sup}_{\Gamma}(X) := |\Gamma(X)| \quad (2)$$

(absolute) support of X *in* Γ. An *(all) frequent itemset mining task* is specified by a dataset Γ and a lower bound minsup ⊆ N on support, called *minimum support*, and asks for enumerating all itemsets with support at least min- sup, called *frequent or (frequent) patterns*.

2 Dynamic Blocking

Lets TID_i and TID_j be two transactions in a dense vertical dataset where:

$$j > i . \quad (3)$$

We name the pair of transactions $\{i, j\}$ a *Block* of product *A*, and we show this block in the form of $\{i,j\}_A^k, k = 1,2,..,n$ where *k* is the number of blocks. Consider the situation where at least α percent of a block contains the product *A*. Therefore, we can represent a vertical dataset in the following series:

$$A = \{TID_i, TID_j - TID_k, TID_l - TID_m, TID_n - TID_o, TID_p - TID_q, TID_r\}. \quad (4)$$

In the problem of FIM with the minimum support of *min-sup*, finding the frequency of an n-itemset in a given dataset is desired. Using *dynamic blocking* approach, the problem is reviewed in the following steps:
Let's

$$\bigcap_{m \subset X}^{o \subset I} \{i,j\}_m^o = \{a,b\} . \quad (5)$$

Let $Maximum_n$ and $Minimum_n$ be the maximum and minimum numbers of the transactions in which all of the n products can be purchased, then the so-called values can be calculated in the as follows:

$$Maximum_n = [\frac{\alpha}{100}.(b\text{-}a)]. \quad (6)$$

$$Minimum_n = (Minimum_{n-1} + \frac{\alpha}{100})\text{-}1. \quad (7)$$

$$Minimum_2 = \frac{2\alpha - 100}{100}. \quad (8)$$

The frequency condition is achieved in the condition that

$$Minimum_n > min\text{-}sup. \quad (9)$$

The contradiction term is also proved where

$$Maximum_n < min\text{-}sup. \quad (10)$$

In the condition that the inequalities (9) and (10) are not satisfied, referring to the main dataset will be demanded. Considerably, the number of referring to the main dataset during running the program decreases when the value of α is notably high. Hence, when $\alpha = 1$, there is no need to waste time reading the main dataset, and it should be read just once in the very beginning of running algorithm; however, this task would not be possible all the time since there is no guarantee for such a dataset being fitted in the RAM. To conclude, decreasing the magnitude of α for each block of each product, we can have a dataset the size of which is sparse enough to be handled in the random memory. Fig 1-3 shows the differences between some of the data representations methods which have been utilized in many algorithms in the memory they used. Furthermore, we tried to show that how the memory usage varies as a function of alpha and bias in dynamic blocking in Fig 4-6.

In the case of referring to the main dataset, it would be wise to save the time utilizing prime number dataset in the main memory and SQL program.

3 IO

The searching time of an itemset within a fixed size dataset increases linearly by the size of the itemset. Nevertheless, in the case that the size of the itemset is variable, it changes exponentially in respect to the size of the itemset. In Maker algorithm, thanks to dynamic blocking approach, the size of the data structure is independent from minimum support. Hence, searching an itemset within the data structure in Maker will not be effected by the low values of minimum support.

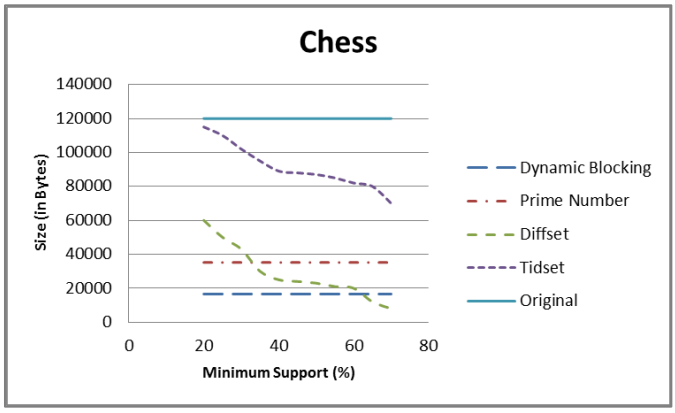


Figure 1. Main Memory Used by the Algorithms in Chess Dataset

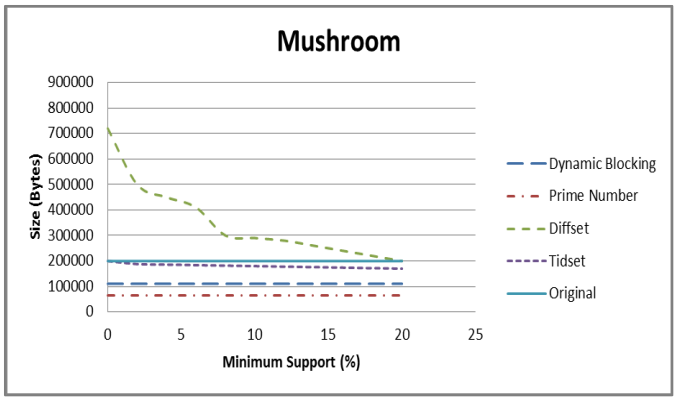


Figure 2. Main Memory Used by the Algorithms in Mushroom Dataset

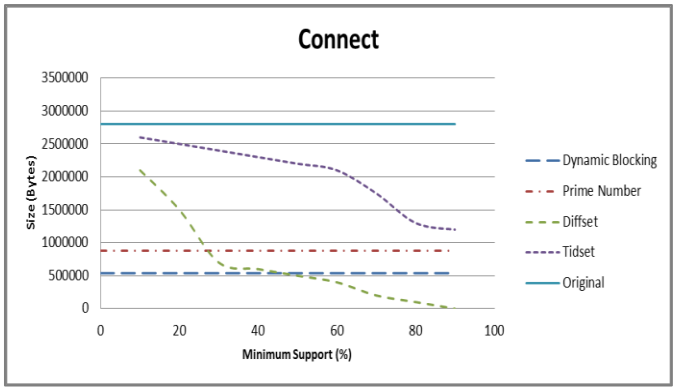


Figure 3. Main Memory Used by the Algorithms in Connect Dataset

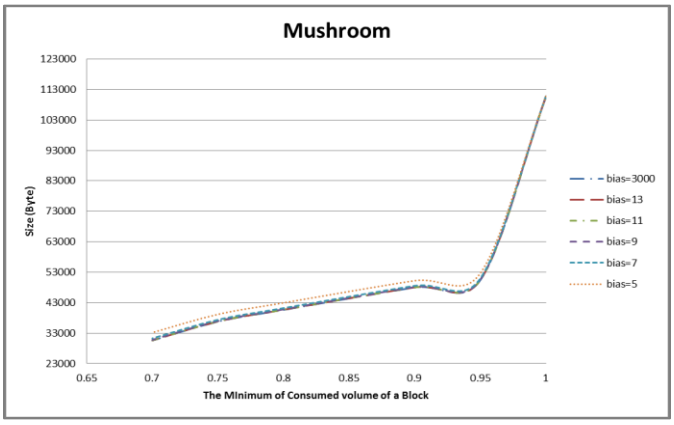


Figure 4. Main Memory Used by the Dynamic Blocking as the Functions of Alpha and Bias in Mushroom Dataset

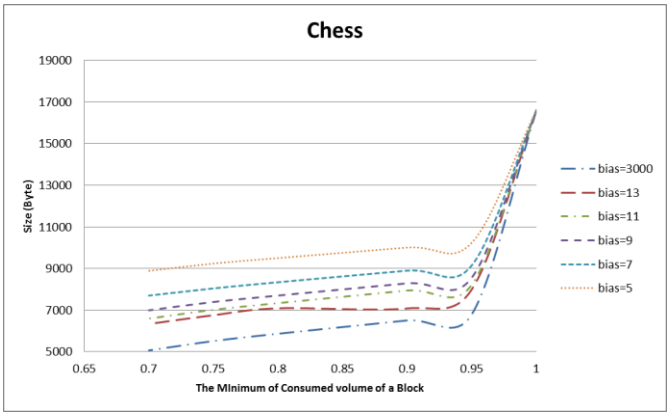


Figure 5. Main Memory Used by the Dynamic Blocking as the Functions of Alpha and Bias in Chess Dataset

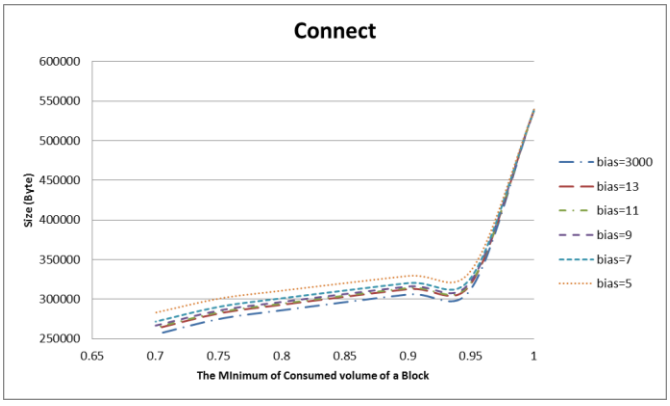


Figure 6. Main Memory Used by the Dynamic Blocking as the Functions of Alpha and Bias in Connect Dataset

4 Experiment and Conclusion

The magnitude of α of the algorithm used in the experiments was zero, and it means that may be in some blocks, the expected product did not exist. All the conditions in the experiments were the same as *FIMI 04*. The dense datasets chosen for the experiments were Chess and Mushroom which were downloaded from http://fimi.ua.ac.be/data/ . As it was expected, in the case that the minimum support is high, Maker is not the best algorithm, and Eclat-1st could perform better. Nevertheless, in the assumed condition, Maker could reach the best performance, and outperformed the best algorithms In *FIMI 04.*

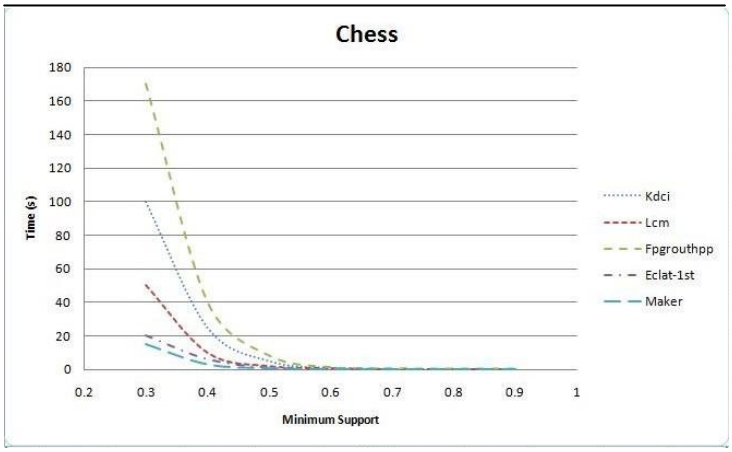


Figure 7. Comparative Performance: Chess Dataset

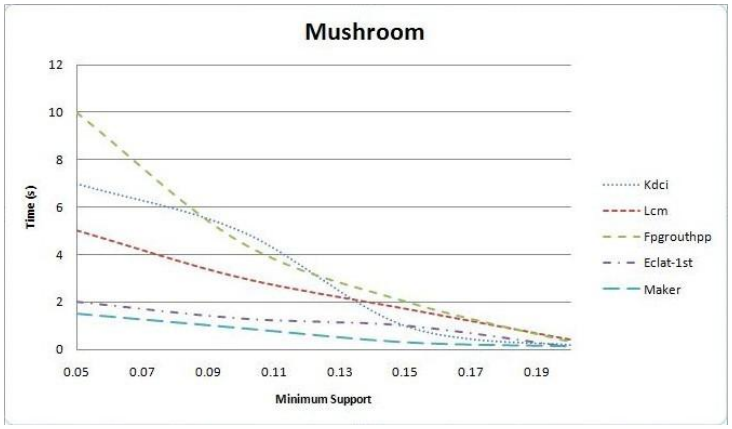


Figure 8. Comparative Performance: Mushroom Dataset

5 References

[1] R. Agrawal, R. Srikant. "Fast algorithms for mining association rules". J. Bocca, M. Jarke, C. Zaniolo, Proceedings of the 20th International Conference on Very Large Data Bases (VLDB'94), Santiago de Chile, September 12-15, pages 487–499. Morgan Kaufmann, (1994).

[2] C. Borgelt." Efficient implementations of Apriori and Eclat". In Goethals and Zaki [8].

[3] S. Orlando, C. Lucchese, P. Palmerini, R. Perego, F. Silvestri." kdci: a multi-strategy algorithm for mining frequent sets". In Goethals and Zaki [8].

[4] T. Uno, T. Asai, Y. Uchida, H. Arimura. “LCM: An efficient algorithm for enumerating frequent closed item sets”. In Goethals and Zaki [8].

[5] G. Sta Grahne, J. Zhu. “Efficiently using prefix-trees in mining frequent itemsets”. In Goethals and Zaki [8].

[6] M. J. Zaki. ”Scalable algorithms for association mining”. IEEE Transactions on Knowledge and Data Engineering, 12(3):372–390, (2000).

[7] Pietracaprina, D. Zandolin. “Mining frequent itemsets using patricia tries”. In Goethals and Zaki [8].

[8] B. Goethals and M. J. Zaki, editors. “Proceedings of the IEEE ICDM Workshop on Frequent Itemset MiningImplementations”, Melbourne, Florida, USA, November 19, (2003).

[9] Lars Schmidt-Thieme, editors. “Algorithmic features of Eclat”, Proceedings of the Workshop on Frequent Itemset Mining Implementations, Helsinki, Finland, (2003).

[10] M.Jamali, F. Taghi yareh, M. Rahgozar. “Fast algorithm for generating association rules with encoding databases layout”.World Academy of Science, page 4. Engineering and Technology, (2005).

Design of a Predictor for MD5 Based Cryptographic Systems: A TVAC-PSO Based Approach

Sonai Ray[1], Monjur Alam[2], Samrat Ray[2], and Ayatullah Faruk Mollah[3]
[2]R&D Division, Cadence Design System, Noida, India
[3]Dept. of Computer Science and Engineering, Jadavpur University, Kolkata, India
Email IDs: ssonai2003@yahoo.com, {alammonjur, sray, afmollah}@gmail.com

Abstract - *This paper puts forward an efficient and optimized method to determine the probable-colluders in a MD5 based cryptosystem. Along with the conventional MD5 implementing architecture, our scheme employs a predictor control block which takes the message stream from the user, and provides the log-list of the equal length bit-streams that are most likely to produce collisions with the message stream in a real time system. The predictor controller uses Time varying Acceleration Coefficient Particle Swarm optimization (TVAC-PSO) algorithm. Besides proposing the algorithm itself, the paper also surveys the performance of the predictor when employed with different hardware platforms.*

Key words: MD-5, collision, TVAC-PSO

1 Introduction

Over the years, tremendous advancement in the field of computing power has seen the decline of DES or triple DES [1-2], as cryptographic algorithms. Also, with the advent of E-commerce, Internet Banking, the concept of digital signature or message authentication [3-4] has been one of the core issues in the mind of a crypto system designer. Message Digest Algorithm (MD5) [5-9] is one of the most successful and used hashing function employed till date to provide message authenticity. However despite having satisfied the necessary strong and weak collision [10] properties, the MD5 based real time crypto systems are not free from collisions. So over the years, the presence of a predictor which can provide the designer with the list of bit patterns which are capable of colliding with the message bit streams has been incumbent in the conventional MD5 architecture [11]. In a normal MD5 plane the message stream is 512 bit long. So employment of a brutal force method would involve 2512 iterations to get the complete list.

In this paper we have explored a hitherto untouched area of MD5 and TVAC-PSO synergism. Here our proposed predictor scheme employs TVAC- PSO based algorithm, which takes fur few steps to give the designer a comprehensive list of the so-called "probable colluders".

Our paper is organized as follows. In section A we discuss the general architecture of MD5. Section B explores TVAC-PSO based algorithm. In section C we provide the building blocks and the algorithm of our proposed predictor-attached MD5 system. Section D contains all the relevant experimental results and a brief overview of our s/w resources. Section E concludes our paper.

2 General Architecture of MD5

There are three basic modules or blocks in MD5 hashing. They are given as below:

(a) *Padding Block:* The message is initially padded so that its length is congruent to 448 modulo 512. Then a 64-bit representation of the length in bits of the original message is appended to this one, so that the message stream is ultimately 512 bit long.

(b) *Initialization of MD Buffer:* In our case we assigned the buffers with these initial values. A=67452301, B=EFCDAB89, C=98BADCFE, D=10325476.

(c) *Message Processing:* The message is further processed as follows:

CV_0=Initialized Buffer values.
$CV_{q+1}=SUM_{32}[CVq,RF_1(Yq,RF_H(Yq,RF_F(Yq,CVq))))]$
$MD=CV_{L-1}$
where Yq= q-th 512-bit block of the message.
L=Number of blocks in the message.
SUM_{32}= mod-32 addition on each word.
MD= Final hash value.
RFm=Round function using primitive logic function 'm'.
A single execution of MD5 involves 16 steps of operations performed on the initial buffers. Each step is of the following form:
A=B+((A+g(B,C,D)+X[K]+T[i])<<s)
where <<s=circular left shift by s bits.
X[k] = K-th 32 bit word in the q-th 512 bit block of the message.
T[i] =ith 32-bit word in matrix T generated by Sine function.
The compression functions used in MD5 are of the following form:
Here ® Denotes XOR function.

Round	Primitive Function (g)	g(B,C,D)
1	F(B,C,D)	(B^C)V(B'^D)
2	G(B,C,D)	(B^D)V(C^D')
3	H(B,C,D)	B®C®D
4	I (B,C,D)	C® (BVD')

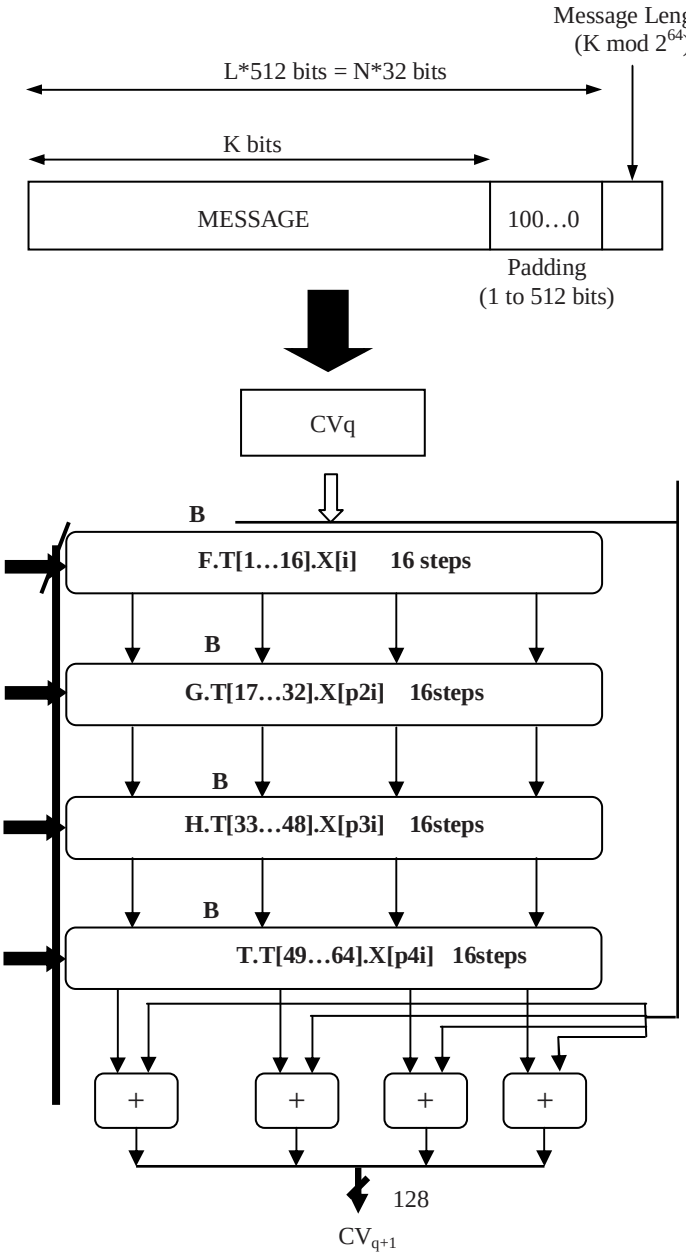


Fig. 1. Basic Modules of MD5

3 TVAC-PSO Based Algorithm

There are a number of population based evolutionary computing techniques such as: Evolution strategies [12-15], genetic algorithms [16-17]. As a new stochastic algorithm PSO [18] and later on TVAC-PSO [19] has enjoyed paramount popularity off late. The time varying acceleration coefficient particle swarm optimization algorithm (TVAC-PSO) is based on the following mathematical relation:

$$Vi_{new}=Vi_{old}+c_1(y_i-x_i)+c_2(y_g-x_i) \quad (1)$$

$$Xi_{new}=xi_{old}+vi_{new} \quad (2)$$

where $C_1=(c_{1f}-c_{1i})*(iter/M)+c_{1i}$ (3)

and $C_1=(c_{2f}-c_{2i})*(iter/M)+c_{2i}$ (4)

With C_1 values gradually decrease from 2.5 to 0.5 and C_2 values gradually increase from 0.5 to 2.5.

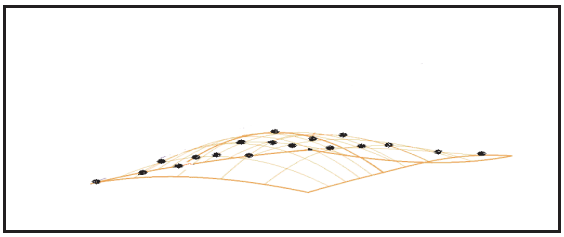


Fig. 2. A Classical representation of PSO. '.' Representing individual swarms in an 'N' dimensional space.

In our proposed scheme,
Xi=The present output of the controller.
Vi=The 512 bit transient vector that needs to be bitwise added with the present x_i
Y_g, y_l denotes global and local maxima. We will discuss them in detail in the following section.
Ci denotes the coefficients in TVAC-PSO. They change their values as per equation 3 and 4. In our scheme, variation of C_1 and C_2 values are chosen to be the classical ones (i.e. between 2.5 to 0.5) to ensure minimum probability of a rush towards false maxima. As mentioned earlier the role of global and local maxima along with a detailed description of our objective function will be presented in the next section.

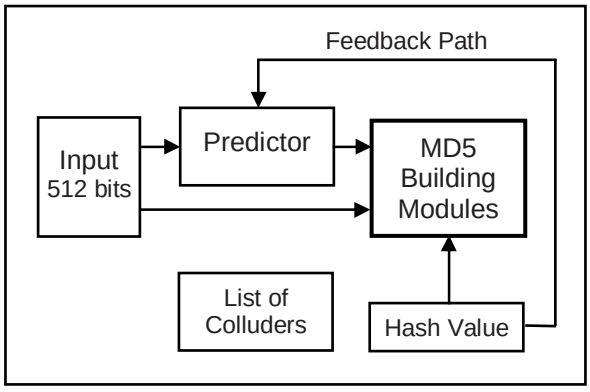


Fig. 3. Our Proposed Predictor Based Block Diagram of MD5

4 Proposed Algorithm for the Predictor

Before we embark upon describing our algorithm, we first provide a brief anecdote of our objective function 'F'.

A formal mathematical representation of our function is given as:

$$F(h(x_i)) =M \quad (5)$$

where, $h(x_i)$ = 128 bit hash value generated by MD5, when input is 512-bit output of the i^{th} swarm.

$F(h(x_i))$ = Decimal equivalent of the 128 bit binary value obtained by bit-wise XNOR-ing $h(x_i)$ with that of the hash value of the original message. M is expressed in normalized scale.

Also Yg =Global maxima, or in other words during any iteration it is that 512-bit binary stream which has recorded maximum 'M' value among all the agents in all the iterations up to that.

Y_l =Local maxima, or in other words during any iteration it is that 512-bit binary stream which has recorded maximum 'M' value among all the agents in that particular iteration.
Now in below we propose our algorithm for the controller of the predictor system.

INPUT: (512 bit Message Stream)
OUTPUT: (Log-List of possible colluders)

START:

1. Create a population of 512 swarms.
2. Initiate each of the swarm content with 512-bit long random values.
3. Apply TVAC_PSO Algorithm.
4. After Each Iteration each of the node content is subjected to optimization function 'F'.
5. The bit stream for which resultant value (M) is greater than a threshold (0.95 here) is noted in a log-list.
6. **if** (Number of colluders<A pre-specified value (here 100))
   GOTO 3
   Else

END

5 Experimental Results

First we provide the performance of our novel predictor based scheme, when implemented in different CPU. We compare our architecture with that of a normal MD5 architecture.

(a) Our Scheme:

CPU	Performance in MB/s	Maximum CPU utilization
ARM7TDMI	2.32	84.4%
ARM9TDMI	2.51	81.0%
ST22	2.66	74.5%
Pentium III	3.10	69.2%

(b) Normal MD5:

CPU	Performance in MB/s	Maximum CPU utilization
ARM7TDMI	2.37	82.1%
ARM9TDMI	2.58	79.6%
ST22	2.71	71.2%
Pentium III	3.12	67.5%

So we can see that employment of our scheme decreases the system speed and increases the CPU utilization to a small extent. But in a trade-off scenario we can overlook them as our scheme can in fact generate the probable colluders list in a surprisingly small number of iteration.

We have tried our scheme with 32 randomly selected 512-bit long message streams. And at the same time we have noted the number of probable colluders it generate after 2^5,2^{10},2^{16} iterations. The detailed statistical however will be made available in the main paer.

Message Number	After 2^5 iterations	After 2^{10} iterations	After 2^{16} iterations
1	3	54	112
2	2	58	116
3	4	61	102
4	1	43	108
5	2	35	107
6	3	57	121
7	5	38	130
8	1	47	110
9	4	55	128
10	6	64	139
11	2	68	142
12	1	49	113
13	3	38	122
14	2	50	127
15	5	42	105
16	2	38	100
17	1	32	134
18	3	31	128
19	4	69	117
20	2	53	112
21	3	65	119
22	5	40	130
23	6	59	142
24	3	38	116
25	2	54	137
26	7	70	102
27	4	54	119
28	3	39	123
29	1	60	135
30	2	53	147
31	5	34	130
32	2	46	112

where the messages used were:

Message Number	Message in Hex
1	04881D0585845DD1F 61E2562242070DB
2	26891E0585845FE3F 64F2593042070DB
3	59572D0785845DD1F 61F2561232070DE

4	23881E068745AD1F 72E2562242070AF
5	07532E3F95845D51F 61F2563212DFE68
6	45982F1542315ED1D 52E4572242070DB
7	14991A0585845ED1A 31F2462143072AC
8	95111F2563745ED1F 41D2562741929AA
9	36541E058584543AA 64F159304390317
10	F7537E82BD31F2352 AD7D2BBEB86D391
11	F4292244432AFF97A B9423A7FC93A039
12	34563287DFEAD2345 1DDEADECB23546A
13	345167DEAC23AACDE AEACABC456367FA
14	F61E2562C040B3406 98D5122455AE905
15	4686C62AD9A2E5514 ADBCFA9289A7FC5
16	154DFEACBEB76819 2673EACBBAFCB41
17	518723539FEABCDEB 9102837BAEACBDD
18	60423D0795545EDCA 42F2561290075AD
19	8B43F6AE6B912219F A579418B6FA5561
20	2381913456ADCEAF2 7612EACD6187AEC
21	9911234165483ADEE ACDEBEAC345127E
22	A9E34563F7523451A 123EACABAEA4123
23	214AECC6FA751ADDA 655B58C4613F02D
24	A4321738ACCBAEC59 572D0785845DD1F
25	BECBADCE234512388 1E068745AD1F451
26	8F0BCBE324171FEAC 21E1CD34289B7FB
27	24982E068745A456 32EE42194207052
28	325BECC6FA751433C 21E2C534279B712
29	3219834165483432A 3219DEA03240335
30	23881E068745AD1F 71E2562242070AF
31	23881E068745AD1F 71E2562242070AF
32	10291E26874565342
	51E55761E20EE32

As we can see the predictor generates a substantial number of potential colluders in a very small number of iterations.

Our Software resources:

For simulation and experimental purpose we used 'C' based pseudo-codes for MD5 and TVAC-PSO algorithms. The standalone PC used by us was a Pentium -4 machine, with a clock speed of 2.4 GHZ and 248 MB of RAM.

6 Conclusion

As we can see from the experimental results and other relevant data, that although incorporation of our predictor controller does hamper the system performance and CPU utilization a bit but, the with only 2^{16} (compared to 2^{512} in case of brute force analysis) iterations the scheme is able to generate a substantial log list (of about 100-150 components) of potential colluders. Thereby within a very small time it gives the user the power to identify the possible bit-streams which can produce close enough hash values to generate potential hazards such as authentication failure in feature.

Although the work is still in progress but it can be expected that in near future predictors which are incorporating stochastic algorithms such as PSO, CPSO, and TVAC-PSO in their controllers will create a whole new avenue in the design of cryptographic hash function.

References

[1] D. Coppersmith. "The Data Encryption Standard (DES) and Its Strength Against Attacks", IBM Journal of Research and Development, May 1994.

[2] H. Heys, C Adams. "Selected Areas in Cryptography", Springer Publications.

[3] R. Atkinson. "Security Architecture for the Internet Protocol", IFTF Network working group.

[4] A. Bosselaers, R. Govaerts, J. Vandewalle, "Fast Hashing on the Pentium", Advances in Cryptography: Crypto-96 Proceeedings, Published by Springer-Verlag.

[5] Hans Dobbertin. "Cryptanalysis of MD5 Compress", 1996, http://citeseer.ist.psu.edu/68442.html.

[6] Xiayoun Wang, Dengguo feng, Xuejia Lai and Hongbo yu, "Collisions for Hash Functions MD4,MD5, Haval-128, and RIPEMD", Cryptology ePrint Archieve-2004.

[7] R. Rivest, "The MD5 Message-Digest Algorithm", Internet RFC 1321, April 1992.

[8] Metzger, P., Simpson, W. "IP Authentication using Keyed MD5", April-1995.

[9] Hefferman, A. "TCP MD5 Signature Option".

[10] Preneel, B., "The State of Cryptographic Hash Functions", Proceedings EURO-CRYPT-96, 1996, Published by Springer-Verlag.

[11] Touch, J., "Report on MD5 Performance", December-1994.

[12] T. Back., "Evolutionary Algorithms in Theory and Practice", Newyork Oxford University Press, 1996.

[13] Beyer, "The Theory of Evolution Strategies", Berlin, Germany, Springer-Verlag, 2001.

[14] I. Rechenberg, J.M. Zurada, R.J. MarksII, C. Robinson, "Evolution Strategy in Computational Intelligence: Imitating Life". IEEE Press, 10994. Page 147-159.

[15] H. Schwefel, "Numerical optimization of Computer models", New York, Wiley, 1981.

[16] D. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning", Wesley-1989.

[17] Z. Michalewicz, "Genetic Algorithms: Data Structure, Evolution Programs", Springer-Verlag 1994.

[18] R.C. Eberhart and J. Kennedy, " A New optimizer using Particle Swarm Theory", In 6th Symposium of Micro Machine and Human Science, Nagoya, Japan, 1995. pp. 39-43.

[19] P.N. Suganthan. "Particle Swarm Optimizer with Neighborhood Operator". In IEEE International Congress on Evolutionary Computation, 1999. Page: 1958-1962.

An Automated Derivation of the Hilbert/Ackerman *Grundzüge* Sentential Calculus from Łukasiewicz's *CN*

Jack K. Horner
P. O. Box 266
Los Alamos, New Mexico 87544 USA

Abstract

Two logics are implicationally equivalent if the axioms and inference rules of each imply the axioms of the other. Characterizing the inferential equivalences of various formulations of the sentential calculi is thus foundational to the study of logic. Using an automated deduction system, I show that the sentential calculus, here called GTL, of Hilbert/Ackerman's Grundzüge der theoretischen Logik, *can be derived from Łukasiewicz's CN. Although each of these systems is known to imply the other, the proof presented here appears to be novel. The automated prover first develops a single set of lemmas and propositions -- an elementary "core" theory -- that constitutes a large fraction of the proofs of five of the six axioms of the "Grundzüge" calculus. In passing, the proofs show that one of the "Grundzüge" axioms can be regarded as a lemma in the proofs of three of the other axioms of that system.*

Keywords: propositional logic, automated deduction, sentential calculus

1.0 Introduction

Two logics are implicationally equivalent if the axioms and inference rules of each imply the axioms of the other. Characterizing the equivalences of various formulations of the sentential calculi is thus foundational to the study of logic ([1],[3]-[7],[9]-[10],[12]-[15]).

"CN", the formulation of the sentential calculus in [1], is among the most compact: its vocabulary contains only two logical connectives (C, and N) and sentence variables (*p, q, r*, ...). It has two inference rules (condensed detachment and substitution), and three axioms.

In CN, any expression of the form *Cxy* or *Nz*, where *x, y,* and *z* are sentences, is a sentence. *Cpq* is interpreted as "sentence *p* implies sentence *q*"; *Np* is interpreted as "not-*p*". N and C are right-associative, and N has higher precedence than C. For example,

CCqrCpNr

translates to the more common "arrow-and-parenthesis" notation as

$$(q \rightarrow r) \rightarrow (p \rightarrow \sim r)$$

where "$\rightarrow$" designates "implies" and "~" designates "not".

The axioms of CN in [1] are:

CN1. *CCpqCCqrCpr*
CN2. *CCNppp*
CN3. *CpCNpq*

Cast in CN notation, the axioms of the Hilbert/Ackerman *Grundzüge* sentential

calculus (GTL, [9]) are (CN theorem identifiers are shown in parentheses):

H1. *CxCyx* (CN18)
H2. *CCxCxyCxy* (CN30)
H3. *CCxCyzCyCxz* (CN21)
H4. *CCyzCCxyCxz* (CN22)
H5. *CpCNpq* (CN3)
H6. *CCpqCCNpqq* (CN54))

The main result of this paper is that the axioms of [1] collectively imply the axioms of [9].

2.0 Method

To show that the axioms of [`] imply the axioms of [9], the *prover9* ([2]) script shown in Figure 1 was executed under on a Dell Inspiron 545 with an Intel Core2 Quad CPU Q8200 @ 2.33 GHz and 8.00 GB RAM, running under the *Windows Vista Home Premium (SP2)/Cygwin* operating environment.

```
set(hyper_resolution).

formulas(usable).
P( i(i(x,y), i(i(y,z), i(x,z))) )      # label("AxCN1").
P( i(i(-x,x), x) )                     # label("AxCN2").
P( i(x, i(-x,y)) )                     # label("AxCN3").
end_of_list.

formulas(sos).
-P(i(x,y)) | -P(x) | P(y)          # label("InfConDet").
end_of_list.

formulas(goals).
P ( i(x, i(y,x)) )                           # label("AxH1").
P ( i(i(x, i(x,y)), i(x,y))  )               # label("AxH2").
P ( i(i(x, i(y,z)), i(y, i(x,z))) )          # label("AxH3").
P ( i(i(y,z), i(i(x,y), i(x,z)))  )          # label("AxH4").
P ( i(x, i(-x,y)) )                          # label("AxH5").
P ( i(i(x,y), i(i(-x,y),y)) )                # label("AxH6").
end_of_list.
```

Figure 1. The *prover9* script (in Horn clause ([11]) form), used to show that the axioms of CN imply the axioms of GTL. The inference rules are condensed detachment and substitution. The implementation of condensed detachment is the formula in the "sos" list; substitution is derived from *prover9*'s hyperresolution rule (introduced in the "set" command at the top of the script). Details of *prover9*'s syntax and semantics can be found in [2].

3.0 Results

Figure 3 shows that GTL implies CN.

```
============================== PROOF =================================

% Proof 1 at 0.01 (+ 0.05) seconds: "AxH5".

5 P(i(x,i(-x,y))) # label("AxH5") # label(non_clause) # label(goal).  [goal].
9 P(i(x,i(-x,y))) # label("AxCN3").  [assumption].
```

```
15 -P(i(c11,i(-c11,c12))) # label("AxH5") # answer("AxH5").  [deny(5)].
16 $F # answer("AxH5").  [resolve(15,a,9,a)].

============================== end of proof ==========================


============================== PROOF =================================

% Proof 2 at 0.11 (+ 0.08) seconds: "AxH1".

1 P(i(x,i(y,x))) # label("AxH1") # label(non_clause) # label(goal).  [goal].
7 P(i(i(x,y),i(i(y,z),i(x,z)))) # label("AxCN1").  [assumption].
8 P(i(i(-x,x),x)) # label("AxCN2").  [assumption].
9 P(i(x,i(-x,y))) # label("AxCN3").  [assumption].
10 -P(i(x,y)) | -P(x) | P(y) # label("InfConDet").  [assumption].
11 -P(i(c1,i(c2,c1))) # label("AxH1") # answer("AxH1").  [deny(1)].
21 P(i(i(i(-x,y),z),i(x,z))).  [hyper(10,a,7,a,b,9,a)].
22 P(i(i(x,y),i(i(-x,x),y))).  [hyper(10,a,7,a,b,8,a)].
23 P(i(i(i(i(x,y),i(z,y)),u),i(i(z,x),u))).  [hyper(10,a,7,a,b,7,a)].
31 P(i(x,x)).  [hyper(10,a,21,a,b,8,a)].
33 P(i(-i(x,x),y)).  [hyper(10,a,9,a,b,31,a)].
35 P(i(i(x,y),i(-i(z,z),y))).  [hyper(10,a,7,a,b,33,a)].
75 P(i(i(x,-y),i(y,i(x,z)))).  [hyper(10,a,23,a,b,21,a)].
98 P(i(x,i(-i(y,y),z))).  [hyper(10,a,21,a,b,35,a)].
104 P(i(i(i(-i(x,x),y),z),i(u,z))).  [hyper(10,a,7,a,b,98,a)].
131 P(i(x,i(y,i(-x,z)))).  [hyper(10,a,21,a,b,75,a)].
146 P(i(i(i(x,i(-y,z)),u),i(y,u))).  [hyper(10,a,7,a,b,131,a)].
341 P(i(x,i(y,y))).  [hyper(10,a,104,a,b,8,a)].
348 P(i(i(i(x,x),y),i(z,y))).  [hyper(10,a,7,a,b,341,a)].
368 P(i(x,i(i(-y,y),y))).  [hyper(10,a,348,a,b,22,a)].
402 P(i(i(i(i(-x,x),x),y),i(z,y))).  [hyper(10,a,7,a,b,368,a)].
867 P(i(i(x,i(-y,y)),i(z,i(x,y)))).  [hyper(10,a,23,a,b,402,a)].
1209 P(i(x,i(y,i(z,x)))).  [hyper(10,a,146,a,b,867,a)].
1235 P(i(x,i(y,i(z,y)))).  [hyper(10,a,867,a,b,1209,a)].
1296 P(i(x,i(y,x))).  [hyper(10,a,1235,a,b,1235,a)].
1297 $F # answer("AxH1").  [resolve(1296,a,11,a)].

============================== end of proof ==========================

============================== PROOF =================================

% Proof 3 at 0.55 (+ 0.09) seconds: "AxH3".

3 P(i(i(x,i(y,z)),i(y,i(x,z)))) # label("AxH3") # label(non_clause) # label(goal).
[goal].
7 P(i(i(x,y),i(i(y,z),i(x,z)))) # label("AxCN1").  [assumption].
8 P(i(i(-x,x),x)) # label("AxCN2").  [assumption].
9 P(i(x,i(-x,y))) # label("AxCN3").  [assumption].
10 -P(i(x,y)) | -P(x) | P(y) # label("InfConDet").  [assumption].
13 -P(i(i(c5,i(c6,c7)),i(c6,i(c5,c7)))) # label("AxH3") # answer("AxH3").  [deny(3)].
21 P(i(i(i(-x,y),z),i(x,z))).  [hyper(10,a,7,a,b,9,a)].
22 P(i(i(x,y),i(i(-x,x),y))).  [hyper(10,a,7,a,b,8,a)].
23 P(i(i(i(i(x,y),i(z,y)),u),i(i(z,x),u))).  [hyper(10,a,7,a,b,7,a)].
31 P(i(x,x)).  [hyper(10,a,21,a,b,8,a)].
33 P(i(-i(x,x),y)).  [hyper(10,a,9,a,b,31,a)].
35 P(i(i(x,y),i(-i(z,z),y))).  [hyper(10,a,7,a,b,33,a)].
69 P(i(i(x,i(y,z)),i(i(u,y),i(x,i(u,z))))).  [hyper(10,a,23,a,b,23,a)].
70 P(i(i(-i(i(i(x,y),i(z,y)),u),i(i(i(x,y),i(z,y)),u)),i(i(z,x),u))).
[hyper(10,a,22,a,b,23,a)].
75 P(i(i(x,-y),i(y,i(x,z)))).  [hyper(10,a,23,a,b,21,a)].
98 P(i(x,i(-i(y,y),z))).  [hyper(10,a,21,a,b,35,a)].
104 P(i(i(i(-i(x,x),y),z),i(u,z))).  [hyper(10,a,7,a,b,98,a)].
131 P(i(x,i(y,i(-x,z)))).  [hyper(10,a,21,a,b,75,a)].
146 P(i(i(i(x,i(-y,z)),u),i(y,u))).  [hyper(10,a,7,a,b,131,a)].
341 P(i(x,i(y,y))).  [hyper(10,a,104,a,b,8,a)].
348 P(i(i(i(x,x),y),i(z,y))).  [hyper(10,a,7,a,b,341,a)].
368 P(i(x,i(i(-y,y),y))).  [hyper(10,a,348,a,b,22,a)].
402 P(i(i(i(i(-x,x),x),y),i(z,y))).  [hyper(10,a,7,a,b,368,a)].
867 P(i(i(x,i(-y,y)),i(z,i(x,y)))).  [hyper(10,a,23,a,b,402,a)].
1202 P(i(x,i(i(y,i(-z,z)),i(y,z)))).  [hyper(10,a,867,a,b,867,a)].
1209 P(i(x,i(y,i(z,x)))).  [hyper(10,a,146,a,b,867,a)].
```

```
1235 P(i(x,i(y,i(z,y)))).  [hyper(10,a,867,a,b,1209,a)].
1296 P(i(x,i(y,x))).  [hyper(10,a,1235,a,b,1235,a)].
1334 P(i(i(i(x,y),z),i(y,z))).  [hyper(10,a,7,a,b,1296,a)].
5360 P(i(i(-x,y),i(i(y,x),x))).  [hyper(10,a,70,a,b,1202,a)].
5457 P(i(x,i(i(x,y),y))).  [hyper(10,a,1334,a,b,5360,a)].
5607 P(i(i(x,i(y,z)),i(y,i(x,z)))).  [hyper(10,a,69,a,b,5457,a)].
5608 $F # answer("AxH3").  [resolve(5607,a,13,a)].

============================== end of proof ==========================

============================== PROOF =================================

% Proof 4 at 0.86 (+ 0.09) seconds: "AxH4".

4 P(i(i(y,z),i(i(x,y),i(x,z)))) # label("AxH4") # label(non_clause) # label(goal).
[goal].
7 P(i(i(x,y),i(i(y,z),i(x,z)))) # label("AxCN1").  [assumption].
8 P(i(i(-x,x),x)) # label("AxCN2").  [assumption].
9 P(i(x,i(-x,y))) # label("AxCN3").  [assumption].
10 -P(i(x,y)) | -P(x) | P(y) # label("InfConDet").  [assumption].
14 -P(i(i(c8,c9),i(i(c10,c8),i(c10,c9)))) # label("AxH4") # answer("AxH4").  [deny(4)].
21 P(i(i(i(-x,y),z),i(x,z))).  [hyper(10,a,7,a,b,9,a)].
22 P(i(i(x,y),i(i(-x,x),y))).  [hyper(10,a,7,a,b,8,a)].
23 P(i(i(i(i(x,y),i(z,y)),u),i(i(z,x),u))).  [hyper(10,a,7,a,b,7,a)].
31 P(i(x,x)).  [hyper(10,a,21,a,b,8,a)].
33 P(i(-i(x,x),y)).  [hyper(10,a,9,a,b,31,a)].
35 P(i(i(x,y),i(-i(z,z),y))).  [hyper(10,a,7,a,b,33,a)].
69 P(i(i(x,i(y,z)),i(i(u,y),i(x,i(u,z))))).  [hyper(10,a,23,a,b,23,a)].
70 P(i(i(-i(i(i(x,y),i(z,y)),u),i(i(i(x,y),i(z,y)),u)),i(i(z,x),u))).
[hyper(10,a,22,a,b,23,a)].
75 P(i(i(x,-y),i(y,i(x,z)))).  [hyper(10,a,23,a,b,21,a)].
98 P(i(x,i(-i(y,y),z))).  [hyper(10,a,21,a,b,35,a)].
104 P(i(i(i(-i(x,x),y),z),i(u,z))).  [hyper(10,a,7,a,b,98,a)].
131 P(i(x,i(y,i(-x,z)))).  [hyper(10,a,21,a,b,75,a)].
146 P(i(i(i(x,i(-y,z)),u),i(y,u))).  [hyper(10,a,7,a,b,131,a)].
341 P(i(x,i(y,y))).  [hyper(10,a,104,a,b,8,a)].
348 P(i(i(i(x,x),y),i(z,y))).  [hyper(10,a,7,a,b,341,a)].
368 P(i(x,i(i(-y,y),y))).  [hyper(10,a,348,a,b,22,a)].
402 P(i(i(i(i(-x,x),x),y),i(z,y))).  [hyper(10,a,7,a,b,368,a)].
867 P(i(i(x,i(-y,y)),i(z,i(x,y)))).  [hyper(10,a,23,a,b,402,a)].
1202 P(i(x,i(i(y,i(-z,z)),i(y,z)))).  [hyper(10,a,867,a,b,867,a)].
1209 P(i(x,i(y,i(z,x)))).  [hyper(10,a,146,a,b,867,a)].
1235 P(i(x,i(y,i(z,y)))).  [hyper(10,a,867,a,b,1209,a)].
1296 P(i(x,i(y,x))).  [hyper(10,a,1235,a,b,1235,a)].
1334 P(i(i(i(x,y),z),i(y,z))).  [hyper(10,a,7,a,b,1296,a)].
5360 P(i(i(-x,y),i(i(y,x),x))).  [hyper(10,a,70,a,b,1202,a)].
5457 P(i(x,i(i(x,y),y))).  [hyper(10,a,1334,a,b,5360,a)].
5607 P(i(i(x,i(y,z)),i(y,i(x,z)))).  [hyper(10,a,69,a,b,5457,a)].
8470 P(i(i(x,y),i(i(z,x),i(z,y)))).  [hyper(10,a,5607,a,b,7,a)].
8471 $F # answer("AxH4").  [resolve(8470,a,14,a)].

============================== end of proof ==========================


============================== PROOF =================================

% Proof 5 at 0.92 (+ 0.09) seconds: "AxH2".

2 P(i(i(x,i(x,y)),i(x,y))) # label("AxH2") # label(non_clause) # label(goal).  [goal].
7 P(i(i(x,y),i(i(y,z),i(x,z)))) # label("AxCN1").  [assumption].
8 P(i(i(-x,x),x)) # label("AxCN2").  [assumption].
9 P(i(x,i(-x,y))) # label("AxCN3").  [assumption].
10 -P(i(x,y)) | -P(x) | P(y) # label("InfConDet").  [assumption].
12 -P(i(i(c3,i(c3,c4)),i(c3,c4))) # label("AxH2") # answer("AxH2").  [deny(2)].
21 P(i(i(i(-x,y),z),i(x,z))).  [hyper(10,a,7,a,b,9,a)].
22 P(i(i(x,y),i(i(-x,x),y))).  [hyper(10,a,7,a,b,8,a)].
23 P(i(i(i(i(x,y),i(z,y)),u),i(i(z,x),u))).  [hyper(10,a,7,a,b,7,a)].
31 P(i(x,x)).  [hyper(10,a,21,a,b,8,a)].
33 P(i(-i(x,x),y)).  [hyper(10,a,9,a,b,31,a)].
35 P(i(i(x,y),i(-i(z,z),y))).  [hyper(10,a,7,a,b,33,a)].
69 P(i(i(x,i(y,z)),i(i(u,y),i(x,i(u,z))))).  [hyper(10,a,23,a,b,23,a)].
```

```
70 P(i(i(-i(i(i(x,y),i(z,y)),u),i(i(i(x,y),i(z,y)),u)),i(i(z,x),u))).
[hyper(10,a,22,a,b,23,a)].
75 P(i(i(x,-y),i(y,i(x,z)))).  [hyper(10,a,23,a,b,21,a)].
98 P(i(x,i(-i(y,y),z))).  [hyper(10,a,21,a,b,35,a)].
104 P(i(i(i(-i(x,x),y),z),i(u,z))).  [hyper(10,a,7,a,b,98,a)].
131 P(i(x,i(y,i(-x,z)))).  [hyper(10,a,21,a,b,75,a)].
146 P(i(i(i(x,i(-y,z)),u),i(y,u))).  [hyper(10,a,7,a,b,131,a)].
341 P(i(x,i(y,y))).  [hyper(10,a,104,a,b,8,a)].
348 P(i(i(i(x,x),y),i(z,y))).  [hyper(10,a,7,a,b,341,a)].
365 P(i(x,i(y,i(-y,z)))).  [hyper(10,a,348,a,b,75,a)].
368 P(i(x,i(i(-y,y),y))).  [hyper(10,a,348,a,b,22,a)].
382 P(i(i(i(x,i(-x,y)),z),i(u,z))).  [hyper(10,a,7,a,b,365,a)].
402 P(i(i(i(i(-x,x),x),y),i(z,y))).  [hyper(10,a,7,a,b,368,a)].
867 P(i(i(x,i(-y,y)),i(z,i(x,y)))).  [hyper(10,a,23,a,b,402,a)].
1202 P(i(x,i(i(y,i(-z,z)),i(y,z)))).  [hyper(10,a,867,a,b,867,a)].
1209 P(i(x,i(y,i(z,x)))).  [hyper(10,a,146,a,b,867,a)].
1235 P(i(x,i(y,i(z,y)))).  [hyper(10,a,867,a,b,1209,a)].
1296 P(i(x,i(y,x))).  [hyper(10,a,1235,a,b,1235,a)].
1314 P(i(i(i(x,i(y,x)),z),i(u,z))).  [hyper(10,a,7,a,b,1235,a)].
1334 P(i(i(i(x,y),z),i(y,z))).  [hyper(10,a,7,a,b,1296,a)].
2680 P(i(x,i(i(i(y,i(z,y)),u),u))).  [hyper(10,a,867,a,b,1314,a)].
4258 P(i(i(i(x,i(y,x)),z),z)).  [hyper(10,a,2680,a,b,2680,a)].
5335 P(i(i(x,i(-y,y)),i(x,y))).  [hyper(10,a,4258,a,b,1202,a)].
5360 P(i(i(-x,y),i(i(y,x),x))).  [hyper(10,a,70,a,b,1202,a)].
5400 P(i(i(i(x,i(-x,y)),z),z)).  [hyper(10,a,5335,a,b,382,a)].
5457 P(i(x,i(i(x,y),y))).  [hyper(10,a,1334,a,b,5360,a)].
5607 P(i(i(x,i(y,z)),i(y,i(x,z)))).  [hyper(10,a,69,a,b,5457,a)].
8363 P(i(-x,i(x,y))).  [hyper(10,a,5400,a,b,5607,a)].
8511 P(i(i(i(x,y),x),x)).  [hyper(10,a,5360,a,b,8363,a)].
8663 P(i(i(x,i(x,y)),i(x,y))).  [hyper(10,a,23,a,b,8511,a)].
8664 $F # answer("AxH2").  [resolve(8663,a,12,a)].

============================== end of proof ==========================

============================== PROOF =================================

% Proof 6 at 14.87 (+ 1.25) seconds: "AxH6".

6 P(i(i(x,y),i(i(-x,y),y))) # label("AxH6") # label(non_clause) # label(goal).  [goal].
7 P(i(i(x,y),i(i(y,z),i(x,z)))) # label("AxCN1").  [assumption].
8 P(i(i(-x,x),x)) # label("AxCN2").  [assumption].
9 P(i(x,i(-x,y))) # label("AxCN3").  [assumption].
10 -P(i(x,y)) | -P(x) | P(y) # label("InfConDet").  [assumption].
17 -P(i(i(c13,c14),i(i(-c13,c14),c14))) # label("AxH6") # answer("AxH6").  [deny(6)].
21 P(i(i(i(-x,y),z),i(x,z))).  [hyper(10,a,7,a,b,9,a)].
22 P(i(i(x,y),i(i(-x,x),y))).  [hyper(10,a,7,a,b,8,a)].
23 P(i(i(i(i(x,y),i(z,y)),u),i(i(z,x),u))).  [hyper(10,a,7,a,b,7,a)].
31 P(i(x,x)).  [hyper(10,a,21,a,b,8,a)].
33 P(i(-i(x,x),y)).  [hyper(10,a,9,a,b,31,a)].
35 P(i(i(x,y),i(-i(z,z),y))).  [hyper(10,a,7,a,b,33,a)].
53 P(i(i(i(i(-x,x),y),z),i(i(x,y),z))).  [hyper(10,a,7,a,b,22,a)].
69 P(i(i(x,i(y,z)),i(i(u,y),i(x,i(u,z))))).  [hyper(10,a,23,a,b,23,a)].
70 P(i(i(-i(i(i(x,y),i(z,y)),u),i(i(i(x,y),i(z,y)),u)),i(i(z,x),u))).
[hyper(10,a,22,a,b,23,a)].
75 P(i(i(x,-y),i(y,i(x,z)))).  [hyper(10,a,23,a,b,21,a)].
98 P(i(x,i(-i(y,y),z))).  [hyper(10,a,21,a,b,35,a)].
104 P(i(i(i(-i(x,x),y),z),i(u,z))).  [hyper(10,a,7,a,b,98,a)].
131 P(i(x,i(y,i(-x,z)))).  [hyper(10,a,21,a,b,75,a)].
146 P(i(i(i(x,i(-y,z)),u),i(y,u))).  [hyper(10,a,7,a,b,131,a)].
341 P(i(x,i(y,y))).  [hyper(10,a,104,a,b,8,a)].
348 P(i(i(i(x,x),y),i(z,y))).  [hyper(10,a,7,a,b,341,a)].
365 P(i(x,i(y,i(-y,z)))).  [hyper(10,a,348,a,b,75,a)].
368 P(i(x,i(i(-y,y),y))).  [hyper(10,a,348,a,b,22,a)].
382 P(i(i(i(x,i(-x,y)),z),i(u,z))).  [hyper(10,a,7,a,b,365,a)].
402 P(i(i(i(i(-x,x),x),y),i(z,y))).  [hyper(10,a,7,a,b,368,a)].
867 P(i(i(x,i(-y,y)),i(z,i(x,y)))).  [hyper(10,a,23,a,b,402,a)].
1202 P(i(x,i(i(y,i(-z,z)),i(y,z)))).  [hyper(10,a,867,a,b,867,a)].
1209 P(i(x,i(y,i(z,x)))).  [hyper(10,a,146,a,b,867,a)].
1235 P(i(x,i(y,i(z,y)))).  [hyper(10,a,867,a,b,1209,a)].
1296 P(i(x,i(y,x))).  [hyper(10,a,1235,a,b,1235,a)].
1314 P(i(i(i(x,i(y,x)),z),i(u,z))).  [hyper(10,a,7,a,b,1235,a)].
```

```
1334 P(i(i(i(x,y),z),i(y,z))).  [hyper(10,a,7,a,b,1296,a)].
2680 P(i(x,i(i(i(y,i(z,y)),u),u))).  [hyper(10,a,867,a,b,1314,a)].
4258 P(i(i(i(x,i(y,x)),z),z)).  [hyper(10,a,2680,a,b,2680,a)].
5335 P(i(i(x,i(-y,y)),i(x,y))).  [hyper(10,a,4258,a,b,1202,a)].
5360 P(i(i(-x,y),i(i(y,x),x))).  [hyper(10,a,70,a,b,1202,a)].
5400 P(i(i(i(x,i(-x,y)),z),z)).  [hyper(10,a,5335,a,b,382,a)].
5457 P(i(x,i(i(x,y),y))).  [hyper(10,a,1334,a,b,5360,a)].
5607 P(i(i(x,i(y,z)),i(y,i(x,z)))).  [hyper(10,a,69,a,b,5457,a)].
8363 P(i(-x,i(x,y))).  [hyper(10,a,5400,a,b,5607,a)].
8511 P(i(i(i(x,y),x),x)).  [hyper(10,a,5360,a,b,8363,a)].
8622 P(i(i(i(i(i(x,y),x),x),z),z)).  [hyper(10,a,5457,a,b,8511,a)].
28289 P(i(i(x,i(i(y,z),y)),i(x,y))).  [hyper(10,a,23,a,b,8622,a)].
40309 P(i(i(i(x,y),z),i(i(z,x),x))).  [hyper(10,a,23,a,b,28289,a)].
43429 P(i(i(x,y),i(i(i(x,z),y),y))).  [hyper(10,a,23,a,b,40309,a)].
47278 P(i(i(i(x,y),z),i(i(x,z),z))).  [hyper(10,a,5607,a,b,43429,a)].
49336 P(i(i(x,y),i(i(-x,y),y))).  [hyper(10,a,53,a,b,47278,a)].
49337 $F # answer("AxH6").  [resolve(49336,a,17,a)].

============================== end of proof ==========================
```

Figure 2. Summary of a *prover9* ([2]) proof showing that CN ([1]) implies GTL ([9]).

The total time to complete the proofs shown in Figure 2 was ~ 19 seconds on the platform described in Section 2.0.

4.0 Conclusions and discussion

Section 3 demonstrates that CN implies GTL. A companion paper ([16]) proves GTL implies CN. The proof in Figure 2 appears to be novel.

The proof of H5 is trivial because it is identical to CN3. The proof s of H1-H4 and H6 are interesting for several reasons. The automated prover first develops a single set of lemmas and propositions -- an elementary "core" theory -- that constitutes a large fraction of the proofs of H1-H4 and H6. In particular, the proofs of H1-H4 and H6 through Line 35 are the same, along the way proving of the *Law of Identity* at Line 31. The proof of H3, furthermore, is reproduced as a subproof in the proofs of H2, H4, and H6 through Line 5607; thus, H3 can be regarded as a lemma for the proofs of H2, H4, and H6. The proof of H2 through Line 8511 (a theorem that appears in C.S. Peirce's logic [17]) is a subproof of the proof of H6, showing that Line 8511 can be regarded as a lemma for the proof of H6.

5.0 References

[1] Łukasiewicz J. *Elements of Mathematical Logic*. Second Edition (1958). Trans. by Wojtasiewicz O. Pergamon Press. 1963.

[2] McCune WW. *prover9 and mace4.* http://www.cs.unm.edu/~mccune/prover9/. 2009.

[3] Aristotle. *Prior Analytics*. Trans. by A. J. Jenkinson. In Aristotle. *The Basic Works of Aristotle*. Ed. by R. McKeon. Random House. 1941. pp. 62-107.

[4] Aristotle. *Posterior Analytics*. Trans. by G. R. G. Mure. In Aristotle. *The Basic Works of Aristotle*. Ed. by R. McKeon. Random House. 1941. pp. 108-186.

[5] Tarski A. *Introduction to Logic*. Trans. by O. Helmer. Dover. 1941.

[6] Hempel C. Studies in the logic of explanation. In Hempel C. *Aspects of Scientific Explanation and Other Essays in*

the Philosophy of Science. Free Press. 1965. pp. 245-290.

[7] Quine WVO. *Philosophy of Logic*. Second Edition. Harvard. 1986.

[8] Chang CC and Keisler HJ. *Model Theory*. North-Holland. 1990.

[9] Hilbert D and Ackerman W. *Grundzüge der theroretischen Logik.* Second Edition. Dover. 1946.

[10] Frege G. Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens. Halle. 1879. Translated in van Heijenoort J. Begriffsschrift, a formula language, modeled upon that of arithmetic, for pure thought. *From Frege to Gödel: A Source Book in Mathematical Logic, 1879-1931*. Harvard. 1967. pp. 3-82.

[11] Horn A. On sentences which are true of direct unions of algebras. *Journal of Symbolic Logic* 16 (1951), 14–21.

[12] Church A. *Introduction to Mathematical Logic.* Volume I. Princeton. 1956.

[13] Birkhoff G and von Neumann J. The logic of quantum mechanics. *Annals of Mathematics* 37 (1936), 823-843.

[14] Kant I. *Kant's Introduction to Logic* (1800). Trans. by Abbott TK. Greenwood Press. 1963.

[15] Cohen MR and Nagel E. *An Introduction to Logic and Scientific Method.* Harcourt, Brace, and Company. 1934.

[16] Horner JK. An automated derivation of Łukasiewicz's CN from the Hilbert/Ackerman "*Grundzüge"* sentential calculus. *Proceedings of the 2011 International Conference on Artificial Intelligence.* CSREA Press. Forthcoming.

[17] Berry, GDW. Peirce's contributions to the logic of statements and quantifiers. In Wiener P and Young F, eds. *Studies in the Philosophy of Charles S. Peirce.* Harvard University Press. 1952.

The Design of Battle Field Robot

D.Bubesh Kumar
Mechanical Engineering Department
Aarupadaiveedu Institute of Technology
Vinayaka Mission University
Kanchipuram ,Tamilnadu,India

Abstract -*The battle field robot is the robot of the future warfare . The future war will be fought with nuclear , chemical and Biological weapons ,where it is difficult for human soldiers to fight the enemies .The battle field robot consists of CCD cameras for eye , servomotors for locomotion , Fuzzy neural network for the brain of the robot . The fuzzy rules are developed and stored in the microprocessor of the robot , and from the microprocessor the fuzzy rules are sent to the neural network ,separate servo motors are placed at the locomotion points of the robot to achieve moments similar to the humans .The fuzzy neural network is trained by the three dimensional multilayered fuzzy neural network , the robot gets signal from the main computers which is placed in the military base .The kinematics of essential movements of the robot are derived with the free body diagram . The training of the neural network is also explained in this paper.*

Keywords:Neural Network, Fuzzy, Humanoid

1 Introduction

In today's modern warfare, many countries are in possession of nuclear , chemical and biological weapons , the soldiers have to work in a dangerous environment . So battle field Robots play a vital role in the battle field. Battle field robots are used in direct combat and in guarding the nuclear power plants and military installations. The robots are designed similar to the Humanoids. This paper deals with locomotion of the robots, the kinematics of the linkages and equations are formed using Euler angles to design the Robot to have movements similar to human movements.

2 Construction of battle field robot

The robot consists of power source usually from battery. CCD cameras are used as the eyes of the robot, the control or the brain of the robot, neural network chip are used. The servo motors which actuates the arms and legs of the robot are controlled by the neural chips The battle field Robot is designed to carry a payload of 100kg of arms and ammunitions rocket launchers and laser guns .Its movements are similar to the bipod robot. It has arms and legs similar to the humanoid. The arms can swivel for 360 degree.
The head of the robot can rotate 360 degree, to enable to identify the object in 3D . The robot has a global positioning system which guides the robot to the required destination .It also has a control box linked to the base control room from which tasks are given in the form of instructions, from the given instructions the battle field robot makes is own decisions and also using the neural chip to achieve its objectives .

3 The essential movements of the robot are designed to qualify for the battle field

1. Walking 2. Jumping 3. Running 4. Combat 5. Shooting 6. Stealth movements 7. Staircase climbing 8. Squatting

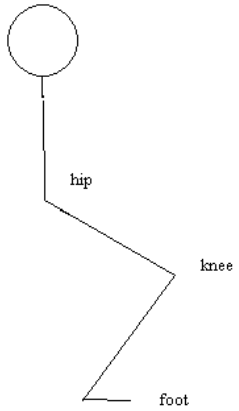


figure 2 Squatting of Battle Field Robot

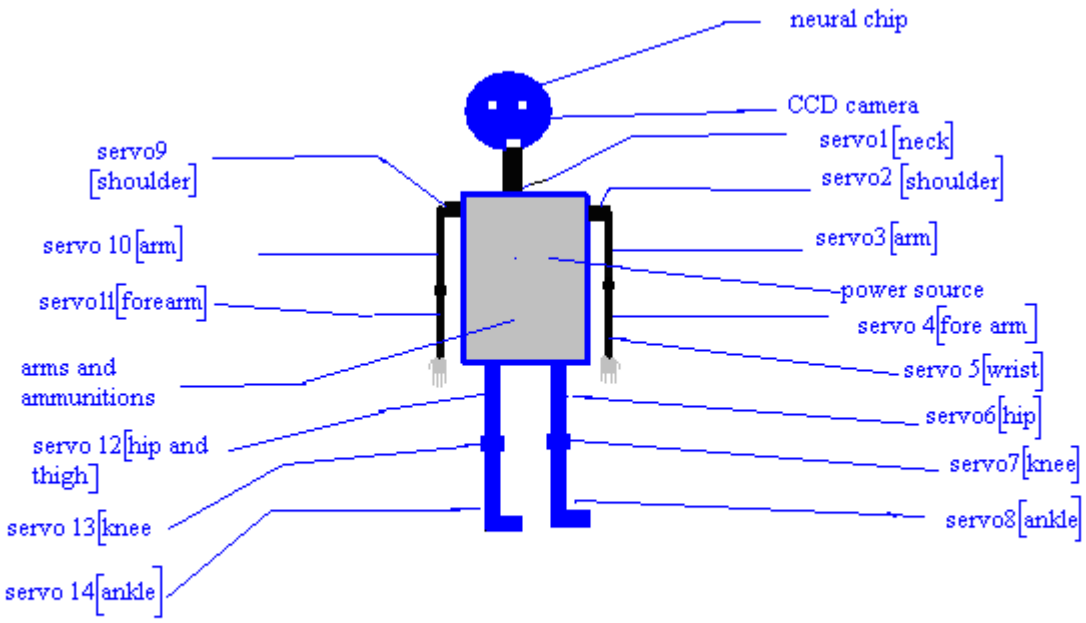


Figure 1 Battle Field Robot

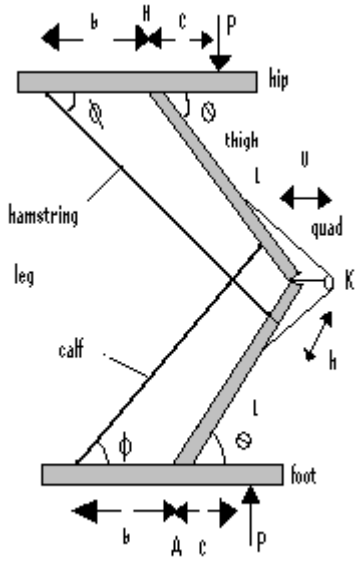


figure 3 FBD of lower portion of battle field robot

4 The squatting performed by a battle field robot

The movement is slow enough to assume static equilibrium a link system is shown in the fig 2 to represent squat. The beam representing hip is connected to the rod representing the upper leg at the hip joint . A tension carrying cord representing the calf muscle (servo 7)connects the foot to the thigh (servo 6) the quad muscle connects the thigh and the leg through a frictionless pulley mechanism representing the patella joint. The joints at H,K and A are hinge joints. The tension in cord (hamstring, calf) and pulley (quads) as a function of the angle the leg makes with the horizontal plane (ɵ)[1].

For simplicity we assume symmetry with respect to the horizontal plane passing through the knee joint .Let P denote the force transmitted at the hip joint to each leg, the force P is the equal to half the weight of the upper body plus the weight of the squat plus the half the weight of the ammunition carried by the battle field robot as shown in figure 3.

Consider the free body diagram of the foot shown in fig 4a.

The movement produced by the force P about A should be equal to the movement produced by the calf muscle at A. (servo 7) as shown in figure 4b.

$$-b\sin\varphi F^{C} + cP = 0 = cP/(b\sin\varphi) \quad (1)$$

$$F^{c} = F^{p} \quad (2)$$

$$(L\cos\theta - c)P - 2d^{k}F^{h}\sin\varphi + d^{q}F^{q} = 0 \quad (3)$$

$$L^{h}\sin\varphi = (L+h)\sin\theta \quad (4)$$

$$L^{h}\cos\varphi = b + (L-h)\cos\theta \quad (5)$$

$$\tan\varphi = (L+h)\sin\theta/[b + (L-h)\cos\theta \quad (6)$$

The ø is expressed in the form of ɵ

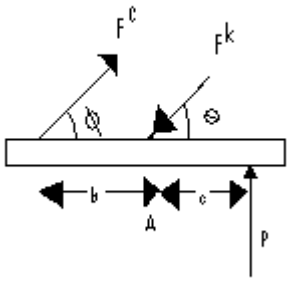


figure 4a FBD of foot

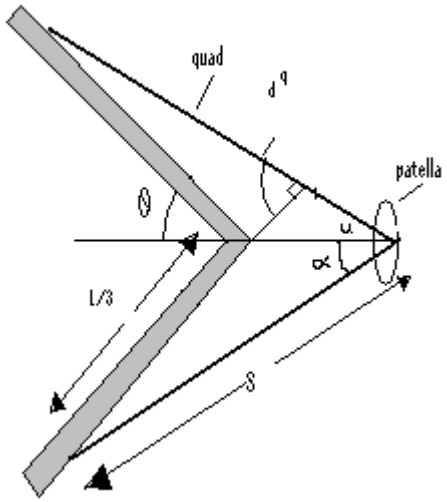


figure 4c FBD Knee joint c

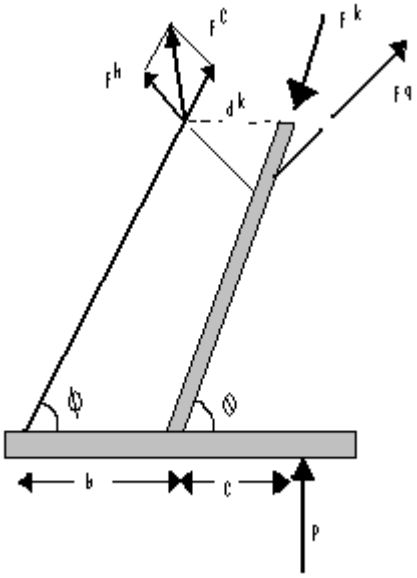


figure 4 b FBD of calf

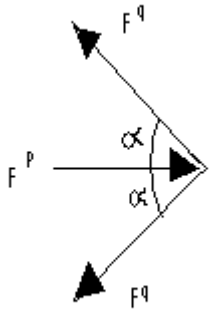


figure 4 d FBD of patella

5 The determination of movement Arm

d^k of the quad muscle group ,the geometry associated with quad muscle is shown in figure 4c.

$$s^2 = (L/3)^2 + u^2 - (2L/3)u\cos(\pi - \theta) \quad (7)$$

$$d^q = u\sin\alpha \quad (8)$$

the compressive force acting on the quads and the knee joint can be found by considering the static equilibrium of K as shown in figure 4d.

$$F^P = 2F^q\cos\alpha \quad (9)$$

6 The vertical jump

Vertical jump performed by battle field robot to reach higher places to leap over obstacles. The propulsive stage of the vertical jump, the battle field root. The segments of the battle field robot change to lift the upper body 70% of the body weight vertically , The mechanic of vertical jumping can be captured with reasonable accuracy by using a four segment model, fig 5a, consists of foot shank, thigh and the upper body. The geometry of the assumed structure dictates that the spatial position of the point mass M at any time t is given by the equation. The figure 5b shows the FBD jumping battle field robot .

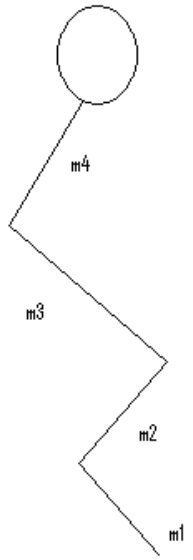


figure 5a jumping robot

$$r = 2L\sin\theta e^2$$

$$v = 2L\cos\theta(d\theta/dt)e_2$$

$$a = [-2L\sin\theta(d\theta/dt)^2 + 2L\cos\theta(d^2\theta/dt^2)]e_2$$

$$P - Mge_2 = M[-2L\sin\theta(d\theta/dt)^2 + 2L\cos\theta(d^2\theta/dt^2)]e_2 \quad (10)$$

$$P = 0 \quad (11)$$

$$g = 2L\sin\theta(d\theta/dt)^2 - 2L\cos\theta(d^2\theta/dt^2) \quad (12)$$

After takeoff

$$a = -ge_2 \quad (13)$$

$$v = (1.625 - gt)e_2 \quad (14)$$

$$r = (1.625t - 0.5gt^2)e_2 \quad (15)$$

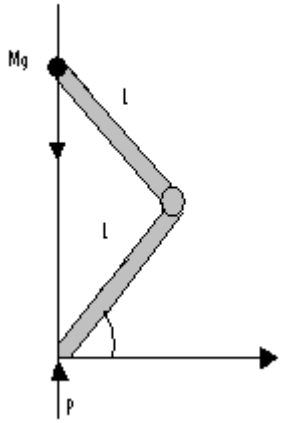


figure 5b FBD jumping battle field robot

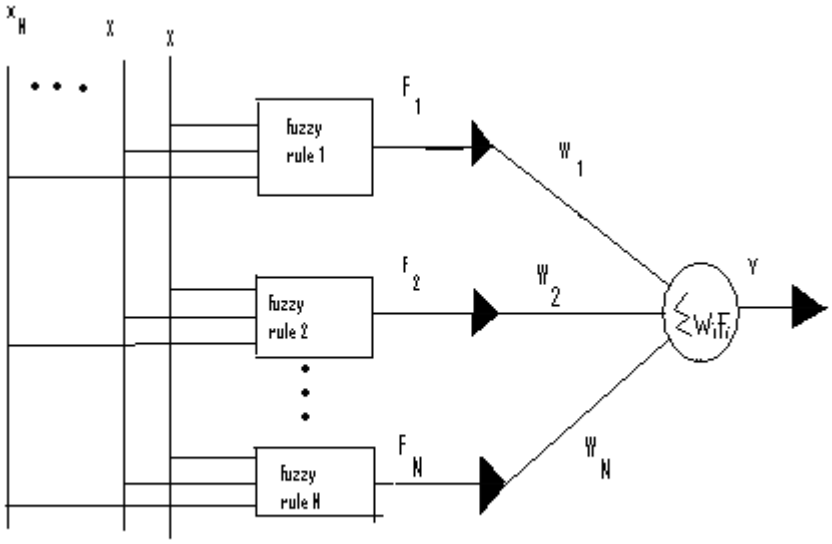


figure 7 Fuzzy Artificial Neural Network

7 Design of neural network

In this paper the humanoid robot control has been designed using fuzzy artificial neural network .Fuzzy neural network fig 7 combine the best of both worlds . They consider fuzzy inputs and at the same time , they are capable of learning .Here the forward network is fuzzy and the feed back is neural network[2] . The neural network fig 8 receives the inputs and the actual outputs, it creates a new classifications and input output associations and it generates new rules. It updates the forward network with new rules.

8 Conclusions

The battle field robot involves more research in the flied of neural network and kinematics of human joints. In this paper one layered network was used to train the robot in the initial phase and in the subsequent phases more layers will be involved in the learning process , to achieve more human like thinking and movements .

9 References

[1] Human Body Dynamics :Classical Mechanics and Human Movement ,Aydin Tozeren , Springer,2000
[2] Understanding Neural Networks ad Fuzzy Logic ,Stamatoos V. Karatalopoulous,IEEE Press,2004

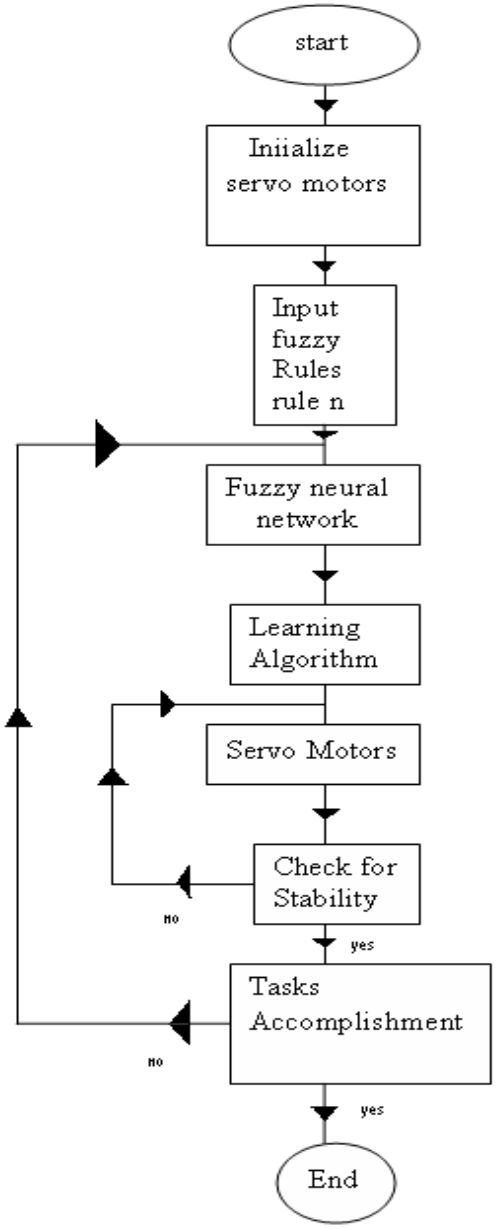


figure 8 flow chart for Fuzzy Neural Network for Battle Field Robot

SESSION

KNOWLEDGE + INFORMATION ENGINEERING + RECOGNITION SYSTEMS + RETRIEVAL METHODS + SEARCH TECHNOLOGIES + EXPERT SYSTEMS

Chair(s)

TBA

Recombinant Knowledge Relativity Threads for Contextual Knowledge Storage

Dr. James A. Crowder
Raytheon Intelligence and Information Systems
16800 E. Centretech Parkway, Aurora, Colorado 80011

Dr. John Carbone
Raytheon Intelligence and Information Systems
1200 S. Jupiter Rd., Garland, Texax 75042

Abstract - *Research shows that generating new knowledge is accomplished via natural human means: mental insights, scientific inquiry process, sensing, actions, and experiences, while context is information, which characterizes the knowledge and gives it meaning [8]. This knowledge is acquired via scientific research requiring the focused development of an established set of criteria, approaches, designs, and analysis, as inputs into potential solutions. This cross-domain research is more commonplace, made possible by vast arrays of available web based search engines, devices, information content, and tools. Consequently, greater amounts of inadvertent cross-domain information content are exposed to wider audiences. Researchers and others, expecting specific results to queries end up acquiring somewhat ambiguous results and responses broader in scope. Therefore, resulting in a lengthy iterative learning process and query refinement, until sought after knowledge is discovered. This recursive refinement of knowledge and context occurs as user cognitive system interaction, over a period in time, where the granularity of information content results are analyzed, followed by the formation of relationships and related dependencies [17]. Ultimately the knowledge attained from assimilating the information content reaches a threshold of decreased ambiguity and level of understanding, which acts as a catalyst for decision-making, subsequently followed by actionable activity or the realization that a given objective or inference has been attained [4, 5].*

1 Introduction

Renowned fuzzy logic theorist Zadeh [24], described tacit knowledge as world knowledge that humans retain from experiences and education, and concluded that current search engines with their remarkable capabilities do not have the capability of deduction, that is the capability to synthesize answers from bodies of information which reside in various parts of a knowledge base. More specifically Zadeh, describes fuzzy logic as a formalization of human capabilities: the capability to converse, reason and make rational decisions in an environment of imprecision, uncertainty, and incompleteness of information.

Underlying decision-making based on informational inferences is a great concern, for informational ambiguity and the ramifications of erroneous inferences can be catastrophic. Often there can be serious consequences when actions are taken based upon incorrect recommendations and those can influence decision-making before the inaccurate inferences can be detected and/or even corrected. This is particularly a problem in intelligence processing. Underlying the data fusion domain is the challenge of creating actionable knowledge from information content harnessed from an environment of vast, exponentially growing structured and unstructured sources of rich complex interrelated cross-domain data.

This paper addresses the challenge of minimizing ambiguity and fuzziness of understanding in large volumes of complex interrelated information content via integration of two cognition based frameworks. The objective is improving actionable decisions using a Recombinant Knowledge Assimilation (RNA) [7] framework integrated with an Artificial Cognitive Neural Framework (ACNF) [3] to recombine and assimilate knowledge based upon human cognitive processes which are formulated and embedded in a neural network of genetic algorithms and stochastic decision making towards minimizing ambiguity and maximizing clarity.

The RNA derivation provides a mathematical relationship for context between two knowledge objects [21]. Described is the research and development to enhance the contextual development between knowledge objects, referred to as the Recombinant kNowledge Assimilation (RNA), along with the Artificial Cognitive Neural Framework (ACNF) which provides the mechanisms by which we apply additional refinement

concepts and formalism for the modular Decomposition/ Reduction/Association sub-processes provided by the RNA.

2 The Problem of Contextual Knowledge

Newell and Simon [12, 13] developed models of human mental processes and produced General Problem Solver (GPS) to perform "means-end analysis" to solve problems by successively reducing the difference between a present condition and the end goal. GPS organized knowledge into symbolic objects and related contextual information which were systematically stored and compared. Almost a decade later Sternberg [23] described a now well-known paradigm called the Sternberg Paradigm where, observations of participants were taken during experiments to determine how quickly the participants could compare and respond with answers based upon the size and level of understanding of their knowledge organized into numerical sets [18, 19, 20]. Sternberg Paradigm is known for (1) organizing knowledge and modifying context while using a common process for describing the nature of human information processing and (2) human adaptation based upon changes in context. Here we introduce an artificial AI framework to provide an autonomous system for analysis of informational context. Figure 1 illustrates the Artificial Cognitive Neural Framework (ACNF). The three main subsystems within the architecture are the Mediator, the Memory System, and the Cognitive System [2].

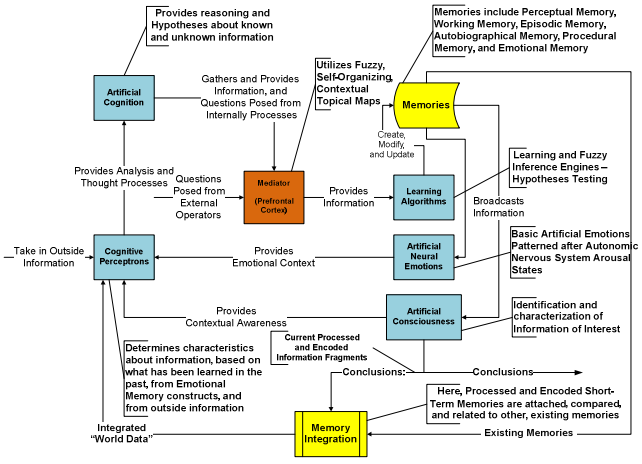


Figure 1 – The Artificial Cognitive Neural Framework

The Mediator gathers information and facilitates communication between agents. Hence, each cognitive decision is handled by the Mediator (the Artificial Prefrontal Cortex) which takes information from perceptrons and from coalitions of perceptrons and updates the short-term, long-term and episodic memories or pedigree [9]. The information available in memory (what the system has learned) is continually broadcast to the conscious perceptrons that form the cognitive center of the system (i.e., they are responsible for the cognitive functionality of perception, consciousness, emotions, processing, etc.) [10]. The purpose of the ACNF is to:

- Provide an architectural framework for "conscious" software agents.
- To provide a "plug-in" domain for the domain-independent portions of the "consciousness" mechanism.
- To provide an easily customizable framework for the domain-specific portions of the "consciousness" mechanism.
- To provide the cognitive mechanisms for behaviors and emotions for "conscious" software agents.

The use of an ACNF for analysis, reasoning, and reporting provides the "Cognitive Intelligence" to allow the top-down executive processing required for real-time cognitive reasoning.

Outlining the need for frameworks which can analyze and process knowledge and context, Liao [25] represented context in a knowledge management framework comprising processes, collection, preprocessing, integration, modeling and representation, enabling the transition from data, information and knowledge to new knowledge [11]. The authors also indicated that newly generated knowledge was stored in a context knowledge base and used by a rule-based context knowledge-matching engine to support decision-making activities. Gupta and Govindarajan [26] defined a theoretical knowledge framework and measured the collected increase of knowledge flow out of multinational corporations based upon "knowledge stock" (e.g., the value placed upon the source of knowledge). Pinto [27] developed a conceptual and methodological framework to represent the quality of knowledge found in abstracts. Suh [28] concluded that collaborative frameworks do not provide the contents which go in them, therefore, content was discipline specific, required subject matter experts, and clear decision making criteria. Additionally, Suh noted that processes promoting positive collaboration and negotiation were required to achieve the best knowledge available, and were characterized by process variables and part of what is defined as the Process Domain. Finally, Ejigu et al. [29] created a framework for knowledge and context which collected and stored knowledge as well as decisions in a knowledge repository that corresponded to a specific context instance. Subsequently, the framework evaluated the knowledge and context via a reasoning engine.

Today, existing databases housing vast bits of information do not store the information content of the reasoning context used to determine their storage [29]. The

knowledge collection and storage formula was therefore developed to include and store relationship context along with knowledge, recursively. This means that, each act of knowledge and context pairing shown as in equation shown in Figure 1 $\sum_{i,j} K_i(R_j)$, recursively examined all of the previous relationships as they were recombined into storage since they were all related and dependent on each other. Recursive refinement then occurred, per iteration of relationship pairing. Recursive refinement occurred when the user found what was looked for shown as $K_i(R_j)$, using interrogatives, (e.g. who, what when, where, why and how) [30, 31]. The information content contributing to finding the answer then has significant value and therefore, a higher degree of permanence in the mind of the stakeholder [32]. Therefore, the information content has reached a threshold where retaining the knowledge and context has become important.

3 Knowledge Relativity Threads

Figure 2 represents a Knowledge Relativity Thread (KRT). This approach for presentation of knowledge and context and was constructed to present five discrete attributes, namely, time, state, relationship distance, relationship value, and event sequence. The goal of a KRT is to map the dependencies of knowledge and related attributes as knowledge is developed from information content. In this figure, the timeline represented by the blue arrow from left to right, shows the events or state transitions in sequence and captures the decision points. During each of the iterations of the presentation of knowledge and context, intrinsic values were captured and placed close to each colored knowledge component. In Figure 2, these are represented as information fragments under the cycles. The Basic Information Decomposition depicts how a KRT looks when it represents information decomposed into pieces; in this case fragments. The red triangles, added next, depict a particular state for each of the iterations, in the KRT development cycle. For emphasis, each colored sphere was built into the depiction and added in sequence to represent the fact that each information fragment follows the other. Each icon represents each information fragment. The relative values in this Basic Knowledge Decomposition between each sphere are perceived to be of the same value to each other. Therefore, the lines are the same distance as well. Since, this base representation depicted in Figure 2 can present time, state, and sequence, as well as, relationships, the challenge was addressed as described by Dourish [33] to create presentation of context which can visually capture and manage a continually renegotiation and redefinition of context as development of knowledge occurs over time.

The KRT depicts cognitive comparison of not just information, but of the contextual relationships also. An important distinction about the observation of each comparison is that each is made from the perspective of the aggregated of information, knowledge, and context.

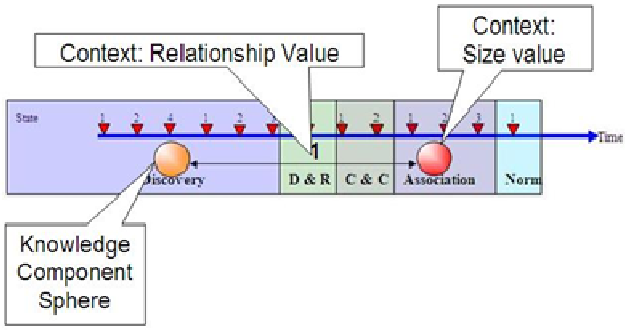


Figure 2– The Knowledge Relativity Thread

The representation of knowledge and context formula is introduced here and is presented by Equation (1). The independent results which follow are mathematical evaluations extended from Newton's law of gravitation shown in Equation (3-1). Newton's Law of Gravitation formula is:

$$F = G\frac{(M_1 M_2)}{r^2} \qquad (1)$$

where:

- F is the magnitude of the gravitational force between the two objects with mass,
- G is the universal gravitational constant,
- M_1 is the mass of the first mass,
- M_2 is the mass of the second mass, and
- r is the distance between the two masses.

This equation was used as an analogy for the derivation of mathematical relationship between a basis, made up of two objects of knowledge [7].

Abstracting Newton's Law of Gravitation as an analogy of Equation (1), representing relationships between two information fragments, using context, is written as Equation (2) shown below, which describes the components of the formula for representing relationships between information fragments using context:

$$A = B\frac{(I_1 I_2)}{c^2} \qquad (2)$$

Where:

- A is the magnitude of the attractive force between the information fragments,
- B is a balance variable,
- I_1 is the importance measure of the first information fragment,
- I_2 is the importance measure of the second information fragment, and
- c is the closeness between the two information fragments.

Comparing the parameters of Equation (1) and Equation (2) F and A have similar connotations except F represents a force between two physical objects of mass M_1 and M_2 and A represents a stakeholder magnitude of attractive force based upon stakeholder determined importance measure factors called I_1, and I_2. As an analogy to F in Equation 1, A's strength or weakness of attraction force was also determined by the magnitude of the value. Hence, the greater the magnitude value, the greater the force of attraction and vice versa. The weighted factors represented the importance of the information fragments to the relationships being formed. The Universal Gravitational Constant G is used to balance gravitational equations based upon the physical units of measurement (e.g. SI units, Planck units). B represents an analogy to G's concept of a balance variable and is referred to as a constant of proportionality. For simplicity, no units of measure were used within Equation (2) and the values for all variables only showed magnitude and don't represent physical properties (e.g. mass, weight) as does G. Therefore, an assumption made here is to set B to the value of 1:

For simplicity, all of these examples assume the same units and B was assumed to be one. The parameter c in Equation (2) is taken to be analogous to r in Equation (1). Stakeholder perceived context known as closeness c represented how closely two knowledge objects (information fragments) (KO) are related. Lines with arrows are used to present the closeness of the relationships between two pieces of knowledge presented as spheroids (see Figure 3).

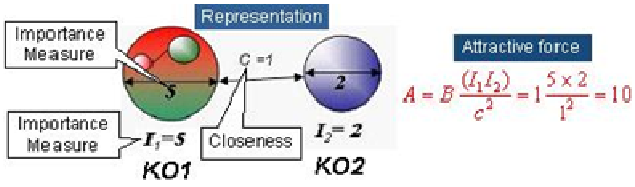


Figure 3– Representation of Knowledge Object and Context

Using Equation (2), the value of the attraction force $A_{1\rightarrow 2}$ = 5 x 2 divided by the relative closeness/perceived distance^2 = 1. Hence, the attraction force A in either direction was 10. The value of 10 is context which can be interpreted in relation to the scale. The largest possible value for attraction force A with the assumed important measure 1-10 scale is 100, therefore a force of attraction value of 10 was relatively small compared to the maximum. This means that the next stakeholder/ researcher understood that a previous stakeholder's conveyance was of small relative overall importance. However, the closeness value of 1 showed that the two objects were very closely related. Figure 4 therefore shows that when using Equation (2), if relationship closeness and/or perceived importance measure of the knowledge objects change value, as new knowledge or context is added and evaluated, then it follows that relationship force of attraction will change.

4 Frameworks for Contextual Knowledge Refinement

As the knowledge and context foundation described above depicts the process and tools for enhancing knowledge and context the Artificial Cognitive Neural Framework expounded upon in the following sections describe the mechanisms by which we apply additional refinement concepts and another formalization for the modular Decomposition and Reduction and Association sub-processes described in the RNA above.

Here we refer again to the ACNF illustrated in Figure 1. The Mediator gathers information and facilitates communication between agents. Hence, each decision handshake of a combined RNA-ACNF system is handled by the Mediator which takes information from perceptrons and from coalitions of perceptrons and updates the short-term, long-term and episodic memories or pedigree. The information available in memory (what the system has learned) is continually broadcast to the conscious perceptrons that form the cognitive center of the system (i.e., they are responsible for the cognitive functionality of perception, consciousness, emotions, processing [14, 15], etc.)

The ACNF contains several different artificial memory systems (including emotional memories) [1, 3], each with specific purposes. Each of these memory systems are stored pedigree used in the recursive RNA process and are integrated during the processes of relationship formation between objects of knowledge and context [6].

When processing pedigree memory, RNA loosely categorizes the granularity of information content into knowledge and context based upon the criteria established by the cognitive human interaction input into the system. These loosely or fuzzy categories are only as fuzzy as the threshold of human understanding. Therefore, in order to artificially create this effect we use Intelligent Software Agents to develop fuzzy organization over time, ultimately reaching a threshold of perceived understanding relative to the initially specified set of criteria.

When processing pedigree memory, RNA loosely categorizes the granularity of information content into knowledge and context based upon the criteria established by the cognitive human interaction input into the system. These loosely or fuzzy categories are only as fuzzy as the threshold of human understanding. Therefore, in order to artificially create this effect we use Intelligent Software

Agents to develop fuzzy organization over time, ultimately reaching a threshold of perceived understanding relative to the initially specified set of criteria.

Illustrated in Figure 4 is an FSSOM with information search hits superimposed. The larger hexagons denote information sources that best fit the search criterion. The isograms denote "closeness"; how close the hits are to particular information topics or criterion.

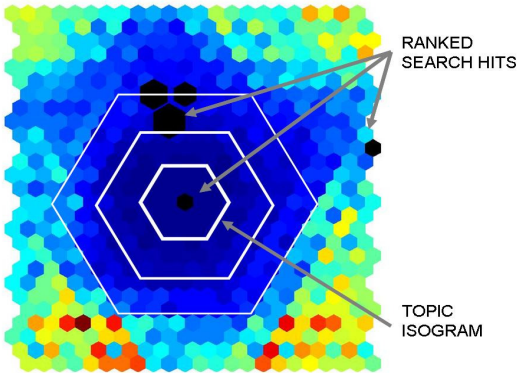


Figure 4– The Fuzzy, Semantic, Self-Organizing Topical Map

There are also other attributes to be explored that would provide significant benefit: as a natural language front end to relational data [6].

Once the FSSOM has been developed, it can be enhanced to include a higher-level Topic Map. This high-level Topic Map describes knowledge structures that span multiple documents. The key features of the Topic Map, illustrated in Figure 5, are the topics, their associations and occurrences in the FSSOM. The topics are the areas on the FSSOM that fall under a topic name. The associations describe the relationships between topics, such as 'biometric data' in 'bone fractures'. The occurrences are the links from the FSSOM into the documents used to form the FSSOM.

5 The Dialectic Search (DS)

The Dialectic Search uses the Toulmin Argument Structure to find and relate information and memories that develops a larger argument, cognitive inference [22]. The Dialectic Search Argument (DSA), illustrated in Figure 6, has four components:

- **Information and Memories**: both in support of and rebutting the argument or hypothesis under analysis by the APC.
- **Warrant and Backing**: explaining and validating the hypothesis.
- **Claim**: defining the hypothesis itself
- **Fuzzy Inference**: relating the information/memories to the hypothesis.

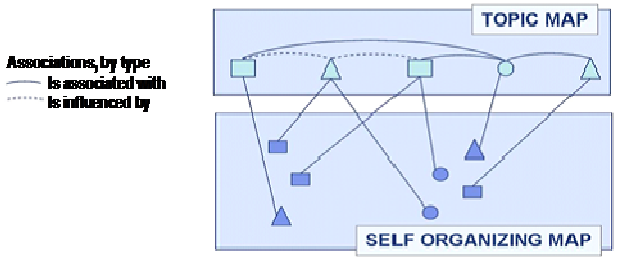


Figure 5– Superimposing High-Level Topical Maps on the FSSOM

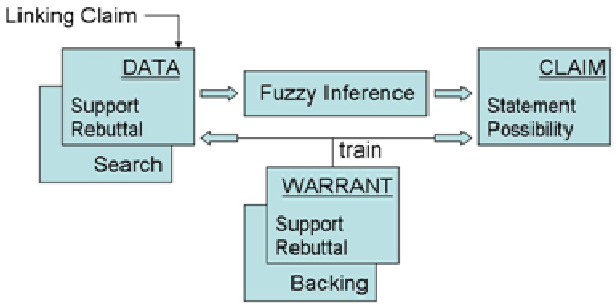


Figure 6– The Dialectic Search Structure

The Dialectic Search serves two purposes:

- First, it provides an effective basis for mimicking human reason.
- Second, it provides a means to glean relevant information from the Topic Map and transform it into actionable cognitive intelligence.

This approach is considered dialectic in that it does not depend on deductive or inductive logic, though these may be included as part of the warrant. Instead, the Dialectic Search depends on non-analytic inferences to find new possibilities based upon warrant examples. The Dialectic Search is dialectic because its reasoning is based upon what is plausible; the Dialectic Search is a hypothesis fabricated from bits of information.

As the Dialectic Search lattice develops, the aggregate possibility is computed using the fuzzy membership values of the support and rebuttal information. Eventually, a Dialectic Search lattice is formed that relates information with its computed possibility. The computation, based on Renyi's entropy theory, uses joint information memberships to generate a robust measure of Possibility, a process that is not possible using Bayesian methods [3].

There is one other valuable attribute to using the

FSSOM method. Because the vector that represents the information is a randomly constructed vector, it cannot be decoded to reformulate the source; the source must be reread. This is critical to protecting compartmentalized information. Using the FSSOM, the protected source can be included in the FSSOM and used to support/rebut an argument without revealing the detailed information.

6 Conclusions and Discussion

As we push to process, analyze and correlate more and more information, the need to combine contextual relevance with information is ever more necessary. When describing how science integrates with information theory, Brillouin [35] defined knowledge succinctly as resulting from a certain amount of thinking and distinct from information which had no value, was the "result of choice," and was the raw material consisting of a mere collection of data. Additionally, Brillouin concluded that a hundred random sentences from a newspaper, or a line of Shakespeare, or even a theorem of Einstein have exactly the same information value. Therefore, information content has "no value" until it has been thought about and thus turned into knowledge.

Information without context is just that, devoid of real content. Instead, the systematic approach presented here, combining the RNA contextual approach, with a cognitive framework, in the ACNF, provides the framework that can handle cognitive processing of information and context, turning them into actionable intelligence. The use of Knowledge Relativity Threads represents the next generation of information analysis and will greatly enhance the capabilities of information processing systems to make sense of increasing volumes multivariate, heterogeneous information [16].

7 References

1. Crowder, J. A., "*Adaptive Learning Algorithms for Functional Constraints on an Evolving Neural Network System*." NSA Technical Paper CON_0013_2002_003 (2002).
2. Crowder, J. A., "*Machine Learning: Intuition (Concept) Learning in Hybrid Genetic/Fuzzy/Neural Systems*." NSA Technical Paper CON_0013_2003_009 (2003).
3. Crowder, J. A., "Operative Information Software Agents (OISA) for Intelligence Processing." AIAA Aerospace@Infotech 2010 Conference.
4. Crowder, J.A., Friess, S., "Artificial Neural Diagnostics and Prognostics: Self-Soothing in Cognitive Systems." International Conference on Artificial Intelligence, ICAI'10 (July 2010).
5. Crowder, J. A., Friess, S., "Artificial Neural Emotions and Emotional Memory." International Conference on Artificial Intelligence, ICAI'10 (July 2010).
6. Crowder, J. A., "Flexible Object Architectures for Hybrid Neural Processing Systems." International Conference on Artificial Intelligence, ICAI'10 (July 2010).
7. Crowder, J. A., Carbone, J, "The Great Migration: Information to Knowledge using Cognition-Based Frameworks." Springer Science, New York (2011)
8. DeYoung, C. G., Hirsh, J. B., Shane, M. S., Papademetris, X., Rajeevan, N., and Gray, J. R. (2010). Testing predictions from personality neuroscience. Psychological Science, 21(6):820-828.
9. T. Gruber, "Collective knowledge systems: Where the social web meets the semantic web," Web Semantics: Science, Services and Agents on the World Wide Web, vol. 6, pp. 4-13, 2008.
10. Marsella, S., and Gratch J., "*A Step Towards Irrationality: Using Emotion to Change Belief*." 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems, Bologna, Italy (July 2002).
11. Miller EK, Freedman DJ, Wallis JD (August 2002). "The prefrontal cortex: categories, concepts and cognition". *Philos. Trans. R. Soc. Lond., B, Biol. Sci.* **357** (1424): 1123–36.
12. Newell, A., "*Unified Theories of Cognition*." Cambridge MA: Harvard University Press (2003).
13. A. Newell, J. C. Shaw, and H. A. Simon, "Preliminary description of general problem solving program-i (gps-i)," WP7, Carnegie Institute of Technology, Pittsburgh, PA, 1957
14. Damasio A (1994) Descarte's error: Emotion, reason, and the human brain. New York: Gosset/Putnam.
15. Davis M, Whalen PJ (2001) The amygdala: vigilance and emotion. Mol Psychiatry 6:13-34.
16. Eichenbaum H (2002) The cognitive neuroscience of memory. New York: Oxford University Press.
17. Kosko, G., "*Fuzzy Cognitive Maps*," International Journal of Man-Machine Studies, 24: 65-75.
18. LaBar KS and Cabeza (2006) Cognitive neuroscience of emotional memory. Nat Rev Neurosci 7: 54-64.
19. LeDoux JE (1996) The Emotional Brain. New York: Simon and Schuster.
20. LeDoux JE (2000) Emotion circuits in the brain. Annu Rev Neurosci 23:155-184.
21. LeDoux JE (2002) Synaptic Self: How our brains become who we are. New York: Viking.
22. Levine, P., "Walking the Tiger: Healing Trauma." North Atlantic Books, Berkeley, CA (1997).
23. J. Rowley and R. Hartley, Organizing knowledge: An introduction to managing access to information: Ashgate Pub Co, 2008.
24. L. Zadeh, "A note on web intelligence, world knowledge and fuzzy logic," Data & Knowledge Engineering, vol. 50, pp. 291-304, 2004

25. S. S. Liao, J. W. He, and T. H. Tang, "A framework for context information management," Journal of Information Science, vol. 30, pp. 528-539, 2004.
26. K. Gupta and V. Govindarajan, "Knowledge flows within multinational corporations," Strategic management journal, vol. 21, pp. 473-496, 2000.
27. M. Pinto, "A grounded theory on abstracts quality: Weighting variables and attributes," Scientometrics, vol. 69, pp. 213-226, 2006.
28. N. Suh, "Application of axiomatic design to engineering collaboration and negotiation," in 4th International Conference on Axiomatic Design, Firenze, 2006.
29. D. Ejigu, M. Scuturici, and L. Brunie, "Hybrid approach to collaborative context-aware service platform for pervasive computing," Journal of computers, vol. 3, pp. 40, 2008.
30. J. I. Hong and J. A. Landay, "An infrastructure approach to context-aware computing," Human-Computer Interaction, vol. 16, pp. 287-303, 2001.
31. N. Howard and A. Qusaibaty, "Network-centric information policy," in The Second International Conference on Informatics and Systems, 2004.
32. J. Anderson, Cognitive psychology and its implications: John r. Anderson: Worth Pub, 2004.
33. J P. Dourish, "What we talk about when we talk about context," Personal and ubiquitous computing, vol. 8, pp. 19-30, 2004
34. J. Rowley and R. Hartley, Organizing knowledge: An introduction to managing access to information: Ashgate Pub Co, 2008.
35. L. Brillouin, Science and information theory: Dover, 2004.

Knowledge Density Mapping for Derivation of Inference Potential

Dr. James A. Crowder

Raytheon Intelligence and Information Systems
16800 E. Centretech Parkway, Aurora, Colorado 80011

Abstract - *Presented is a mathematical derivation and development of an information processing system's Inference Potential. This Inference Potential is determined from providing a measure of the Knowledge Density and Analytical Competency of the information processing systems, based on the contextual assessment of the question, or topic posed by the operator and analyst. The use of Knowledge Density and Analytical Competency to determine an AI systems Inference Potential will provide the methodologies to radically improve the performance and quality of Intelligence Processing Systems by allowing the system to "self analyze" their ability to answer questions and perform the analysis asked of them by operators and analysts.*

1 Introduction

The underlying issues and challenges posed by the introduction of Artificial Intelligence into system designs are not new. Information processing and dissemination systems are an expensive infrastructure to operate and more-often-than-not, these systems fail to provide analysts with tangible and useful situational information, typically overwhelming information analysts with system messages and other low-level data. Real-time human decision making processes must be supported by information derived from the fusion process and must operate in a uniform and cooperative model, fusing data into information and knowledge so information analysts can make informed decisions. One such construct that would aid the information analyst would be a measure of a system's ability to provide quality information and/or inference about a particular subject or question posted by the information analyst. Described here is the mathematical derivation and development of an information processing system's ***Inference Potential***. This *Inference Potential* is determined from providing a measure of the ***Knowledge Density*** and ***Analytical Competency*** of the information processing systems, based on the contextual assessment of the question, or topic posed by the operator and analyst. Such a measure would allow analysts to quickly understand the system's ability to provide quality knowledge about a subject, question, or topic, and could be used to discover knowledge holes or gaps in information processing systems. ***Knowledge Density Mapping*** facilitates information, intelligence, and memory integration, and allows faster accommodation of knowledge and knowledge characteristics. The *Analytical Competency* measure provides analysis, reasoning, and reporting capabilities of an Information Processing System's capabilities (provides cognitive intelligence).

2 Knowledge Density Mapping: A Pathway to AI Metacognition

As we push for "autonomous" systems, the need to provide a system with the ability to understand its own limitations and capabilities and to reason about them, in light of the duties or missions it is given, is becoming increasingly necessary. In humans, we call this ability "Metacognition." Metacognition in humans refers to higher order thinking which involves active control over the cognitive processes engaged in learning and performing. Activities such as planning how to approach a given task, monitoring comprehension, and evaluating progress toward the completion of a task are metacognitive in nature [1]. In an AI system, Metacognition, or ***Knowledge of Cognition***, refers to what a system knows about its own cognition or about cognition in general. In short, it describes the system's ability to think about how and what it thinks. It includes three different kinds of metacognitive awareness: declarative, procedural, and conditional knowledge.

- **Declarative Knowledge**: refers to knowing "about" things,
- **Procedural Knowledge**: refers to knowing "how" to do things, and
- **Conditional Knowledge**: refers to knowing the "why" and "when" aspects of cognition.

We can classify Knowledge of Cognition into three components [2]:

- **Metacognitive Knowledge**: (also called metacognitive awareness) is what the system knows about itself as a cognitive processor [13].
- **Metacognitive Regulation**: is the regulation of cognition and learning experiences through a set of activities that help the system control its learning [14]. This may be based on its understanding of its own "knowledge gaps."
- **Metacognitive Experiences**: are those experiences that have something to do with the current, on-going cognitive endeavors (current mission).

The push to define metacognition within an AI system drives us toward defining an overall Cognitive Ontology to allow metacognitive concepts to be defined within the context of an AI cognitive framework [3]. This need for metacognitive concepts within a system stems from the understanding that knowledge advances not by copying reality but by schematizing it within a formal framework. This allows emergent behavior to be recognized and captured [12]. An **emergent behavior** might be the formation of a new concept, 'bubbling up' from below the artificial conscious level of the system [4]. A simple way of stating this is that the systems would preserve their own **attention** and would, at every level, be concerned with avoiding *interruption and distraction* from their tasks or missions. Figure 1 illustrates the Artificial Cognitive Neural Framework created to accommodate Intelligent Information Agents that provide Metacognitive capabilities. Figure 2 provides the Metacognition Cognitive Lower Ontology for the Intelligent Information Agents [5].

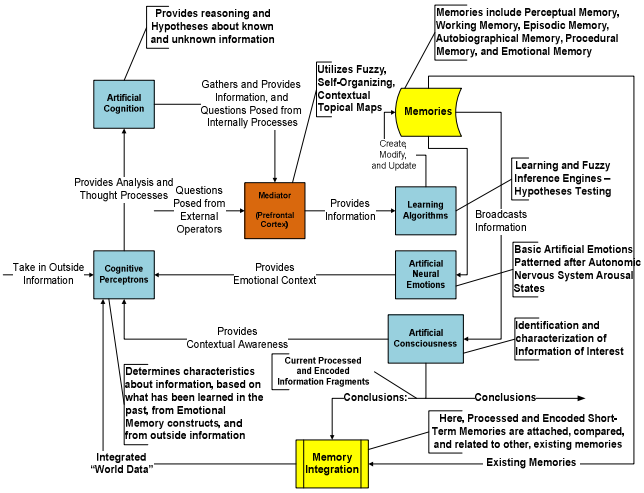


Figure 1 – Artificial Cognitive Neural Framework

In order to achieve metacognitive abilities within the overall AI system, the system must have the ability to measure its own knowledge about a particular topic or subject [6]. This measure of topical or subject knowledge involves measuring the "***density***" of knowledge the system possesses about this subject or topic in question. This ***Knowledge Density*** measure is based on the number of separate information fragments relative to the taxonomy of the topic or subject. Figure 3 provides the Knowledge Density Measure, based on separable topical information fragments [7]. In order to provide the parameters required to compute Knowledge Density, cognitive maps [15] track separable information fragments by topic, as illustrated in Figure 4.

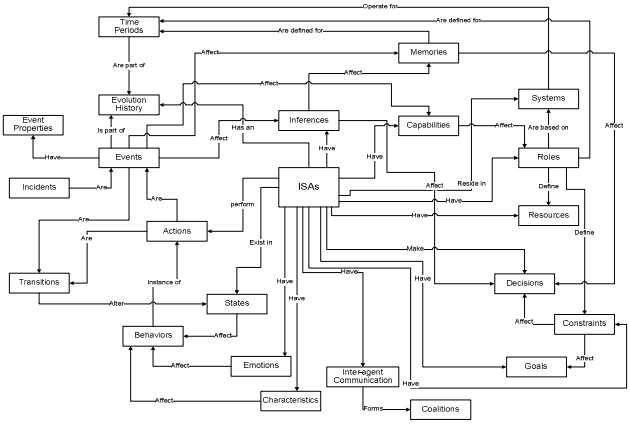


Figure 2 – The ACNF Cognitive Lower Ontology

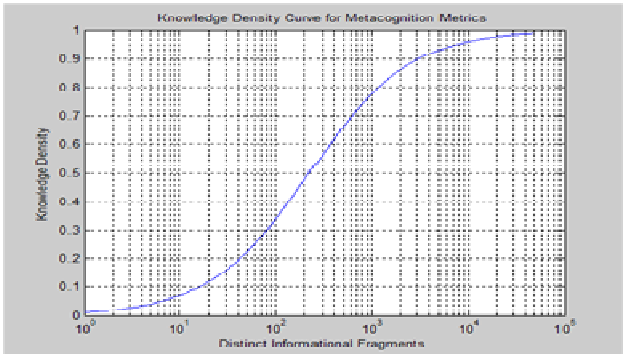


Figure 3 – Knowledge Density Computation

We use knowledge fragment measurements to ensure that we only store information relative to a topic or subject once. Information that is taken in is parsed and information fragments that have not been stored before are pulled out and stored in a cognitive map for that topic. Renyi's entropy measurement is utilized to separate information into topical information fragments. Renyi's entropy measurement is defined as [8]:

$$H_R(Y) = -\log \int_y p(y)^2 \, dy$$

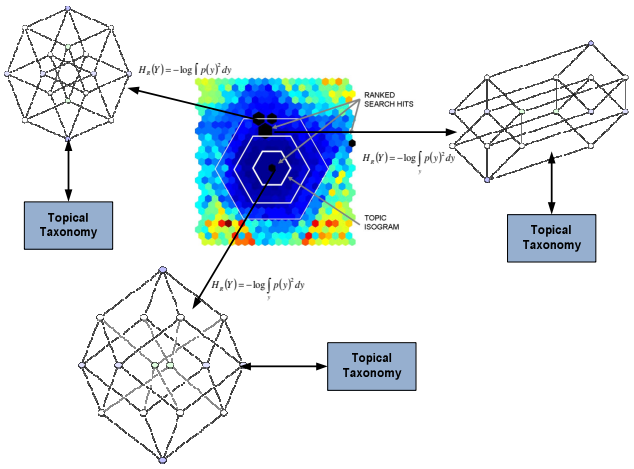


Figure 4 – Knowledge Density Mapping

Computationally, this is difficult, however, Renyi's measure, combined with the Parzen Density estimation method provides a computational model. We start by looking at the information densities, $p(y)$, as a sum of related topical cognitive maps, each centered at y_i, we get:

$$p(y) = \frac{1}{N}\sum_{i=1}^{N} G(y - y_i, \sigma I)$$

Therefore, Renyi's entropy can be computed as the sum of local information interactions (separate information fragments) over all pairs of informational entities. Informational associations are created within the Cognitive Topical Maps utilizing a ***Fuzzy Possibilistic Network*** and Inference Engine, based on Renyi's Theoretics. We use this possibilistic network because:

- It's robust in the presence of inexact information.
- It utilizes conditional possibilistics
  - Mutual Information measurement
  - Joint Informational membership rather than joint probabilities
- Excellent at showing qualitative relationships not attainable with Bayesian methods
  - Excellent at showing qualitative relationships not attainable with Bayesian methods
  - Creates decisions with conditional possibilistic attributes
- More useful with general questions about a subject domain

This methodology allows the Cognitive Topical Maps to be populated with separable information fragments, relative to a topic that maps to the topical taxonomy. This allows a measurement of the density of knowledge a system contains, relative to a topic or subject. Fuzzy, Self-Organizing Contextual Topical Maps are used to measure topics and how other topics relate. Knowledge Density is a measure of the density of knowledge a system has about a topic and the density of related topics that would be used to answer questions and/or analyze situations. The next piece of the Inference Potential computation is Analytical Competence, or, what is the competency of the system to provide the analysis being asked.

3 Analytical Competency

In order to quantifiably measure the a system's ***Inference Potential***, the system must be able to assess its ability to analyze information relative to a question or mission posed to it. We call this measure of analytical potential ***Analytical Competency***. The Analytical Competency measures relative to a topic or subject are based on the algorithms and software that is available:

- The algorithms actual technical skills - what is was designed to do
- The algorithms experiences – tied to emotional memory [16]
- The algorithms body of knowledge – what it has learned

Analytical Competency is tied to "***Areas of Expertise***" within the AI system. Figure 5 illustrates the information flow for the ***Analytical Competency*** measure.

One main element of the overall Analytical Competency measure is a measure of the algorithm's experiences, i.e., what have the algorithms processed before and what has been right and wrong with the analytical output. This is measured utilizing "Emotional Memories" within the AI system [17]. Within the ACNF framework, Drives, Priorities, and Constraints influence emotions. The behavioral subsystem receives situations and computes actions, while memories provide personality parameters and the various conscious agents' sensitivities to emotional computation [18]. If the cross-connectivity of the neural layers is considered as a matrix, we can compute emotional response from the column-wise fuzzy weightings (based on Dr. Levine's Autonomic Nervous System States) and the action response from the row-wise fuzzy weightings [7, 20]. This is analogous to the amygdale and hippocampus that involved in implicit and explicit emotional memories within the human brain [19].

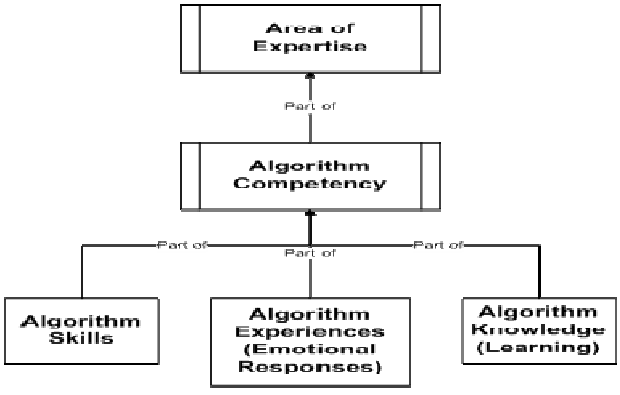


Figure 5 – Analytical Competency Measure Model

Respectively, the ACNF and the cognitive perceptron coalitions become emotionally aroused when they form semantic and episodic memories about situations that cause

"stress" within an artificial neural system. Stress situations may involve a loss of resources, new data environments that are unfamiliar, new interfaces that are introduced into the environment or situations where the algorithms produced incorrect results. These cognitive representations of emotional situations better referred to as memories about emotions rather than emotional memories.

The effects of emotional arousal on explicit memory are due to processes that are secondary to the activation of emotional processing systems in the ACNF [9]. These emotional responses or emotional memories within the algorithmic long-term memories provide vital information that relates to how these algorithms have been able to respond or not respond to given assignments, topical analysis, or missions that have been assigned to the system.

Activity in these areas would be detected by the cognitive coalitions and would lead to increases in system emotional arousal (due to activation of modulation within the neural structure that leads to the release of cognitive problem, solution, search, and emotion agents [10]. These responses are stored and utilized, in part, as a measure of the analytical competency of a set of algorithms that make up an area of expertise within the system. The transmittal of informational content as well as emotional context allows information retrieval performance to be greatly enhanced, allowing for "cognitive economy" within the artificial neural system [11]. The ***Analytical Competency*** measure is based on inputs to the areas shown in Figure 5, illustrated in Figure 6 [18].

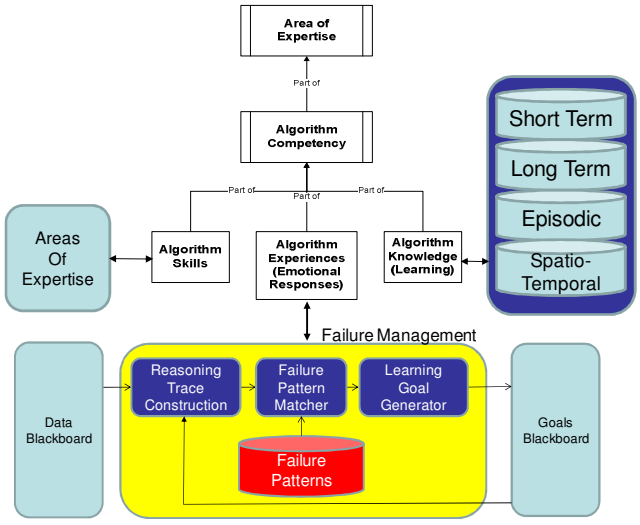


Figure 6 – Analytical Competency Measure Inputs

The actual Analytical Competency measurement is computed as:

$$AC_t = \frac{\sum_{i=1}^{n} w_i \sqrt{L_i^2 + A_i^2}}{n}, \text{ where}$$

$$w_i = \sqrt{\frac{\sum_{j=1}^{m} (\pm) E_j^2}{m}}, \text{ where } E_j = \text{emotional memory response}$$

$$L_i = \frac{\sum_{k=1}^{p} A_{E_k} C_k}{p}, \text{ where } A_{E_k} = \text{memories for area of expertise k and}$$

$$C_k = \text{completeness of memory}$$

$$A_i = \frac{\sum_{l=1}^{r} \mathrm{A}_{\mathrm{R}_l}}{r}, \text{ where } \mathrm{A}_{\mathrm{R}_l} = \text{Algorithm relevancy for lth algorithm}$$

The result is a rating from 0 to 1 of the Analytical Competency of the system for the question or mission posed.

4 Conclusions and Discussion

Once the Knowledge Density and Analytical Competency have been computed, the overall Inference Potential of the system for a given topic/subject/mission is:

$$IP = KD * AC$$

Producing a number between 0 and 1, where 0 means the system has no potential to produce a useful inference for the topic requested and 1 indicates that not only can the system produce useful inferences, but that the inferences will be useful and trustworthy.

Much more research is needed to validate this work and produce an automated way to compute Knowledge Density and Analytical competency. This work is also dependent on further research work on Artificial Neural Emotions, Metcognitive and Metamemory constructs, as well as further work on the ACNF. The purpose of this work is to provide a framework as research continues, for autonomous AI system to provide meaningful knowledge and self-assessments for operators of these systems.

5 References

1. Crowder, J. A., "*Integrating an Expert System into a Neural Network with Genetic Programming for Process Planning*." NSA Technical Paper TIT_01_01_013_2001_001 (2001).
2. Crowder, J. A. "*ELYSE, The Evolving, Life-like Yielding, Symbiotic Environment.*" NSA Technical Paper Technical Paper TIT_01_01_013_2001_002 (2001).
3. Crowder, J. A., "*Adaptive Learning Algorithms for Functional Constraints on an Evolving Neural Network System*." NSA Technical Paper CON_0013_2002_003 (2002).
4. Crowder, J. A., "*Machine Learning: Intuition (Concept) Learning in Hybrid Genetic/Fuzzy/Neural Systems*." NSA Technical Paper CON_0013_2003_009 (2003).
5. Crowder, J. A., "Operative Information Software Agents (OISA) for Intelligence Processing." AIAA Aerospace@Infotech 2010 Conference.
6. Crowder, J.A., Friess, S., "Artificial Neural Diagnostics and Prognostics: Self-Soothing in

Cognitive Systems." International Conference on Artificial Intelligence, ICAI'10 (July 2010).
7. Crowder, J. A., Friess, S., "Artificial Neural Emotions and Emotional Memory." International Conference on Artificial Intelligence, ICAI'10 (July 2010).
8. Crowder, J. A., "Flexible Object Architectures for Hybrid Neural Processing Systems." International Conference on Artificial Intelligence, ICAI'10 (July 2010).
9. Crowder, J. A., Carbone, J, "The Great Migration: Information to Knowledge using Cognition-Based Frameworks." Springer Science, New York (2011)
10. Marsella, S., and Gratch J., "*A Step Towards Irrationality: Using Emotion to Change Belief*." 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems, Bologna, Italy (July 2002).
11. Newell, A., "*Unified Theories of Cognition*." Cambridge MA: Harvard University Press (2003).
12. Damasio A (1994) Descarte's error: Emotion, reason, and the human brain. New York: Gosset/Putnam.
13. Davis M, Whalen PJ (2001) The amygdala: vigilance and emotion. Mol Psychiatry 6:13-34.
14. Eichenbaum H (2002) The cognitive neuroscience of memory. New York: Oxford University Press.
15. Kosko, G., "*Fuzzy Cognitive Maps*," International Journal of Man-Machine Studies, 24: 65-75.
16. LaBar KS and Cabeza (2006) Cognitive neuroscience of emotional memory. Nat Rev Neurosci 7: 54-64.
17. LeDoux JE (1996) The Emotional Brain. New York: Simon and Schuster.
18. LeDoux JE (2000) Emotion circuits in the brain. Annu Rev Neurosci 23:155-184.
19. LeDoux JE (2002) Synaptic Self: How our brains become who we are. New York: Viking.
20. Levine, P., "Walking the Tiger: Healing Trauma." North Atlantic Books, Berkeley, CA (1997).

Heuristic for Simulation Checking

Antonella Santone

Dipartimento di Ingegneria, University of Sannio, Via Traiano 1, 82100 Benevento, Italy

santone@unisannio.it

ABSTRACT

Equivalence checking is a common problem in formal software design. It is the process of determining whether two systems are equivalent to each other according to some mathematically defined notion of equivalence. In this paper to improve efficiency we present a procedure, based on heuristic searches, for checking the well-know simulation relation defined by Milner. We use heuristic mechanisms for the exploration of the search space, in order to avoid the construction of the complete state graph.

Keywords: heuristic searches, equivalence checking, formal methods.

1. INTRODUCTION

Process algebra represents a mathematically rigorous framework for modelling concurrent systems of interacting processes. There are many examples of Process Algebra. The Calculus of Communicating Systems (CCS) of Milner [21] is one of the well known process algebra. Two techniques for verification are provided: *equivalence checking* and *model checking*.

In the *model checking* technique [6] one formulates certain wanted properties in a temporal logic. Each property is checked for validity against the system.

Equivalence checking is the process of determining whether two systems are equivalent to each other according to some mathematically defined notion of equivalence. Equivalence checking can be used to check if a system design conforms to its high-level "service specification". For example, if the observable behavior of a communications protocol is identical to that of a perfect communication channel that delivers all messages in order, then it would be justifiable to deem the protocol correct.

Two algorithmic families can be considered to perform the equivalence checking. The first one is based on refinement principle: *given an initial partition, find the coarsest partition stable with respect to the transition relation* see for example the algorithm proposed by Paige and Tarjan in [23]. The other family of algorithms is based on a cartesian product traversal from the initial state [10, 14]. These algorithms are both applied on the whole state graph, and they require an explicit enumeration of this state space. This approach leads to the well-known state explosion problem. A possible solution is to reduce the state graph before performing the check as shown in [9] where symbolic representation of the state space is used.

Several approaches have been developed to solve the state explosion problem. They are general methods that typically are used to reduce the state space explosion while verifying a set of properties Among them, reduction techniques based on process equivalences [4, 15], symbolic model checking techniques [20], on-the-fly techniques [17], local model checking approaches [27], partial order techniques [11, 13, 25, 28], and abstraction approaches [5].

Before we describe our approach, let us briefly review *heuristic search* [24] which is one of the classical techniques in Artificial Intelligence and has been applied to a wide range of problem-solving tasks including puzzles, two player games and path finding problems. A key assumption of heuristic search is that we can assign a *utility* or *cost* to each state. This cost guides the search suggesting the next state to expand; in this way the most promising paths are considered first.

Two well-known heuristic search algorithms for solving state-space search problems are A* and AO* [19, 24]. A* finds a solution that takes the form of a sequence of operators leading from a start node to a goal node. AO* finds a solution that has a conditional structure and takes the form of a tree, or more generally, an acyclic graph. When a heuristic function is admissible (i.e., the distance to the goal is never overestimated), A* and AO* will find optimal solutions.

In this paper first we formalise the simulation checking as a state-space search problem; then we present an efficient simulation checking procedure, based on heuristic search. We define an admissible function for verifying CCS processes equivalence. This heuristics is syntactically defined, i.e., based on the CCS specifications only, and it is automatically computed, thus there is no need for user intervention or manual efforts. In this paper we apply the AO* algorithm with our admissible heuristic function. This approach extends traditional techniques to efficiently explore the search

Paper submitted to ICAI 2011.

space. The heuristic overcomes the bottleneck of the exhaustive exploration of the global state graph of the two systems to be proved that one simulates the other one.

As far as we know, it is the first attempt to exploit process algebra-based heuristics for equivalence checking in concurrent systems.

The paper is organised as follows: in Section 2 the AO* heuristic search strategy is briefly recalled. Section 3 is a review of the basic concepts of CCS and simulation, while Section 4 presents our approach. Finally, we discuss related work and conclude in Section 5.

2. HEURISTIC SEARCH: AND/OR GRAPHS AND ALGORITHM AO*

Blind search algorithms, such as Breadth-First (BFS) and Depth-First (DFS) searches, are simple and can in principle find solutions to any state space problem. However, they usually result inefficient and impractical in case of large search spaces, as those of most real problems. On the other hand, heuristic search algorithms can be applied to improve search efficiency. Two well-known heuristic search algorithms for solving state-space search problems are A* and AO* [19, 24]. A* finds a solution that takes the form of a sequence of operators leading from a start node to a goal node. AO* finds a solution that has a conditional structure and takes the form of a tree, or more generally, an acyclic graph. In this section we briefly review AND/OR graphs and the heuristic search algorithm AO* for solving problems formalized as AND/OR graphs. The reader can refer to [19] for further details.

A common problem solving strategy consists of decomposing a problem P into subproblems, so that either all or just one of these subproblems need to be solved in order to obtain a solution for P. Representing each problem as a node in a directed graph, where arcs express the decomposition relationship between problems and subproblems, two kinds of nodes arise related to both types of decomposition, AND nodes and OR nodes, and the obtained structure is an AND/OR graph.

More formally, an AND/OR graph G is a directed graph with a special node s, called the *start* (or *root*) *node*, and a nonempty set of *terminal leaf nodes* denoted as $t, t_1, \ldots$. The start node s represents the given problem to be solved, while the terminal leaf nodes correspond to subproblems with known solutions. The nonterminal nodes of G are of three types: OR, AND, and *nonterminal leaf*. An OR node is solved if at least one of its immediate subproblems is solved, while an AND node is solved only when every one of its immediate subproblems is solved. A nonterminal leaf node has no successors and is unsolvable. AND nodes are recognized from OR nodes by connecting the subproblems arcs by a line.

Given an AND/OR graph G, a solution of G is represented by an AND/OR subgraph, called *solution graph of* G, having the following properties: it contains the start node, all its terminal nodes are solved, and if it contains an AND node, it must also contain all the immediate successors.

We assume a cost function assigns a cost to each arc. Each directed arc (m, n) in a graph has a discrete arc cost $c(m, n) > 0$. We also assume each goal node has a cost of zero. Let D be a solution graph and n be a node in D, the cost of n, denoted $h(n)$ is defined as follows:

1. $h(n) = 0$ if n is a terminal leaf node.
2. $h(n) = \infty$ if n is a nonterminal leaf node.
3. $h(n) = c(n, p) + h(p)$ if n is an OR node and it exists p, the immediate successor of n.
4. $h(n) = \sum_{i=1}^{k}[c(n, p_i)+h(p_i)]$ if n is an AND node with immediate successors $p_1, p_2, \ldots, p_k$.

A solution graph of an AND/OR graph G is a *minimal-cost solution graph* if the cost of its root is equal to the minimum over the cost of the root of all the solution graphs of G. The objective of any heuristic search algorithm for AND/OR graphs is to find a minimal-cost solution graph; the algorithm uses the heuristic estimate function $\widehat{h}$, which can be viewed as an estimate of h, to direct the search and to restrict the number of nodes expanded with acceptable limits. Thus, the heuristic search can find an optimal solution graph without evaluating the entire state space. Since in most domains the AND/OR graph G is unknown in advance, it is not supplied explicitly to a search algorithm. We refer to G as the *implicit graph*; it is specified implicitly by a start node s and a successor function. The search algorithm works on an *explicit graph* G', which initially consists of the start node s. The start node is then expanded, that is, all the immediate successors of s are added to the explicit graph G'. At any moment, the explicit graph has a number of *open nodes*, which are nodes with no successors in the explicit graph, and the search algorithm chooses one of these open nodes for expansion. In this manner, more and more nodes and arcs get added to the explicit graph, until finally it has one or more solution graphs as subgraphs. One of these solution graphs is then output by the search algorithm. Table 1 outlines the algorithm AO* for finding the minimal-cost solution graph in an AND/OR graph.

For a more precise description of the algorithm the reader can refer to [19].

An important property holds: AO* returns a minimal-cost solution graph if the heuristic estimate function $\widehat{h}$ satisfies the so-called *admissibility* condition, i.e. $\widehat{h}$ is optimistic. More formally:

DEFINITION 2.1 (ADMISSIBILITY). *A heuristic estimate function $\widehat{h}$ defined on the nodes of an AND/OR graph G is admissible if for each node n in G,*

$$\widehat{h}(n) \leq h(n).$$

3. THE CALCULUS OF COMMUNICATING SYSTEMS

Let us now briefly recall the Calculus of Communicating Systems (CCS) [21], which is an algebra suitable for modelling and analysing processes. The reader can refer to [21] for further details. The syntax of *processes* is the following:

$$p ::= nil \mid \alpha.p \mid p + p \mid \ p|p \ \mid p\backslash L \mid p[f] \mid x$$

where α ranges over a finite set of actions $\mathcal{A}ct = \{\tau, a, \overline{a}, b, \overline{b}, ...\}$. Input actions are labeled with "non-barred" names, e.g. a,

1. Initialize the graph to start node.
2. Traverse the graph following the current path accumulating nodes that have not yet been expanded or solved.
3. Pick any of these nodes and expand it and if it has no successors set the $\widehat{h}$ value of this node to FUTILITY (equivalent to an "infinite cost") otherwise calculate $\widehat{h}$ for each of the successors.
4. If $\widehat{h}$ is 0 then mark the node as SOLVED.
5. Change the value of $\widehat{h}$ for the newly created node to reflect its successors by back propagation.
6. Wherever possible use the most promising routes and if an AND (or OR) node is marked as SOLVED then mark the parent node as SOLVED.
7. If starting node is SOLVED or value greater than FUTILITY, stop, else repeat from 2.

Table 1: Sketch of the AO* algorithm.

while output actions are "barred", e.g. $\overline{a}$. The action $\tau \in \mathcal{A}ct$ is called *internal action*. The set L, in processes with the form $p\backslash L$, ranges over sets of *visible actions* ($\mathcal{V} = \mathcal{A}ct - \{\tau\}$), f ranges over functions from actions to actions, while x ranges over a set of *constant* names: each constant x is defined by a constant definition $x \stackrel{\text{def}}{=} p$. Given $L \subseteq \mathcal{V}$, with $\overline{L}$ we denote the set $\{\overline{l} \mid l \in L\}$. We call $\mathcal{P}$ the processes generated by p.

The standard *operational semantics* [21] is given by a relation $\longrightarrow \subseteq \mathcal{P} \times \mathcal{A}ct \times \mathcal{P}$, which is the least relation defined by the rules in Table 2 (we omit the symmetric rule of **Sum** and **Par**).

A *(labeled) transition system* is a quadruple $(\mathcal{S}, \mathcal{A}ct, \longrightarrow, p)$, where $\mathcal{S}$ is a set of states, $\mathcal{A}ct$ is a set of transition labels (actions), $p \in \mathcal{S}$ is the initial state, and $\longrightarrow \subseteq \mathcal{S} \times \mathcal{A}ct \times \mathcal{S}$ is the transition relation. If $(p, \alpha, q) \in \longrightarrow$, we write $p \xrightarrow{\alpha} q$.

If $\delta \in \mathcal{A}ct^*$ and $\delta = \alpha_1 \ldots \alpha_n, n \geq 1$, we write $p \xrightarrow{\delta} q$ to mean $p \xrightarrow{\alpha_1} \cdots \xrightarrow{\alpha_n} q$. Moreover $p \xrightarrow{\lambda} p$, where λ is the empty sequence. Given $p \in \mathcal{S}$, with $\mathcal{R}(p) = \{q \mid p \xrightarrow{\delta} q\}$ we denote the set of the states reachable from p by $\longrightarrow$.

Given a CCS process p, the *standard transition system* for p is defined as $\mathcal{S}(p) = (\mathcal{R}(p), \mathcal{A}ct, \longrightarrow, p)$. Note that, with abuse of notation, we use $\longrightarrow$ for denoting both the operational semantics and the transition relation among the states of the transition system.

Given a process p, $\mathcal{F}irst(p) = \{\alpha \in \mathcal{A} \mid \exists p' \text{ s.t. } p \xrightarrow{\alpha} p'\}$ denotes the set of all the first actions that p can perform. It can be syntactically defined as the least solution of the following recursive definition:

Definition 3.1 (First actions).

$$
\begin{array}{lcl}
\mathcal{F}irst(nil) & = & \emptyset \\
\mathcal{F}irst(\alpha.p) & = & \{\alpha\} \\
\mathcal{F}irst(p+q) & = & \mathcal{F}irst(p) \cup \mathcal{F}irst(q) \\
\mathcal{F}irst(p\backslash L) & = & \mathcal{F}irst(p) - (L \cup \overline{L}) \\
\mathcal{F}irst(x) & = & \mathcal{F}irst(p) \quad \text{if } x \stackrel{\text{def}}{=} p \\
\mathcal{F}irst(p[f]) & = & \{f(\alpha) \mid \alpha \in \mathcal{F}irst(p)\} \\
\mathcal{F}irst(p \mid q) & = & \begin{cases} \mathcal{F}irst(p) \cup \mathcal{F}irst(q) \cup \{\tau\} \\ \qquad \text{if } \exists \alpha \in \mathcal{F}irst(p) \text{ and } \exists \overline{\alpha} \in \mathcal{F}irst(q) \\ \mathcal{F}irst(p) \cup \mathcal{F}irst(q) \\ \qquad \text{otherwise} \end{cases}
\end{array}
$$

3.1 Simulation

The CCS language, like all other process algebras, can be used to describe both implementations of processes and specifications of their expected behaviours. Therefore CCS supports the so-called single language approach to process theory, that is, the approach in which a single language is used to describe both actual processes and their specifications. An important ingredient of these languages is therefore a notion of behavioural equivalence. One process description, say SYS, may describe an implementation, and another, say SPEC, may describe a specification of the expected behaviour. This approach to program verification is also sometimes called implementation verification.

In the following we introduce the well-known notions of simulation. Milner introduces simulation as following.

Definition 3.2. *(simulation).*
Let p and q be two CCS processes and α an action.

- *A simulation, $\mathcal{R}$, is a binary relation on $\mathcal{P} \times \mathcal{P}$ such that $p\mathcal{R}q$ implies:*
  (i) $p \xrightarrow{\alpha} r'$ implies $q \xrightarrow{\alpha} q'$ with $p'\mathcal{R}q'$
- *We say that q simulates p, written $p \preceq q$, iff there is a simulation $\mathcal{R}$ with containing the pair (p, q).*

4. HEURISTIC FOR CHECKING SIMULATION

Simulation checking can be seen as a search problem. Suppose that we want to check whether a process q simulates p, i.e. $p \preceq q$. We formalise this problem as a state-space search problem, as follows:

$$P = (\mathcal{N}, \mathcal{O}, N_0, \mathcal{G}).$$

We now explain each component of the above problem P.

Node ($\mathcal{N}$) and initial node (N_0)

Each node $N \in \mathcal{N}$ is a 4-uple $\langle r, s, n, a \rangle$, where r (resp. s) is a state belonging to the transition system for p (resp. for q), n is 1 when r has to move, while is equal to 2 when it is up to s to move and a is an action or the symbol λ. It is an action β when it is the turn of s to move and s

$$\textbf{Act}\ \frac{}{\alpha.p \xrightarrow{\alpha} p} \qquad \textbf{Sum}\ \frac{p \xrightarrow{\alpha} p'}{p+q \xrightarrow{\alpha} p'} \qquad \textbf{Con}\ \frac{p \xrightarrow{\alpha} p'}{x \xrightarrow{\alpha} p'}\ x \stackrel{\text{def}}{=} p$$

$$\textbf{Par}\ \frac{p \xrightarrow{\alpha} p'}{p|q \xrightarrow{\alpha} p'|q} \qquad \textbf{Com}\ \frac{p \xrightarrow{l} p',\ q \xrightarrow{\bar{l}} q'}{p|q \xrightarrow{\tau} p'|q'} \qquad \textbf{Rel}\ \frac{p \xrightarrow{\alpha} p'}{p[f] \xrightarrow{f(\alpha)} p'[f]}$$

$$\textbf{Res}\ \frac{p \xrightarrow{\alpha} p'}{p\backslash L \xrightarrow{\alpha} p'\backslash L}\ \alpha \notin (L \cup \overline{L})$$

Table 2: Operational semantics of CCS.

has to perform the action β, i.e. the same action that r has previously performed. a is equal to λ when it is the turn of r to move. The initial node is of the form: $\langle p, q, 1, \lambda\rangle$ meaning that it is the turn of p to move and the process q has to simulate p.

Operators ($\mathcal{O}$)

Operators transform one node of $\mathcal{N}$ into another node. The operators are given in Table 3. Formally, the operators are given by a relation $\rightsquigarrow \subseteq \mathcal{N} \times \mathcal{A}ct \times \mathcal{N}$. If $(N, \alpha, N') \in \rightsquigarrow$ we write $N \overset{\alpha}{\rightsquigarrow} N'$. To check whether $p \preceq q$ one has to build an AND/OR graph whose root is labeled with the initial node. The immediate successors of a node are determined by the application of one of the operators in Table 3.

The operator in Table 3 with the form:

$$\frac{\textit{premise}}{N \xrightarrow{n_1} N_1 \text{ and } \cdots \text{ and } N \xrightarrow{n_m} N_m}$$

where *premise* is the antecedent of the rule, generates all the successors of the AND node N. Graphically, the arcs $(N, N_1) \cdots (N, N_m)$ are connected by a line as in Figure 1.

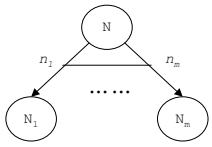


Figure 1: And node.

On the other hand, the operator with the form:

$$\frac{\textit{premise}}{N \xrightarrow{n_1} N_1 \text{ or } \cdots \text{ or } N \xrightarrow{n_m} N_m}$$

generates all the successors of the OR node N, which graphically is represented as in Figure 2.

Now, we informally explain the rules in Table 3. In rule $\mathbf{op}_1$, if p can move performing an action α and reaches the process p' then it is the turn of q to move with the same action α. Thus, from $\langle p, q, 1, \lambda\rangle$ we reach $\langle p', q, 2, \alpha\rangle$. In rule $\mathbf{op}_2$, q must simulate the action α performed by p. Thus, if $q \xrightarrow{\alpha} q'$ from $\langle p, q, 2, \alpha\rangle$ we reach $\langle p, q', 1, \lambda\rangle$, meaning that now it is the turn of p to move. Note that $\langle p, q, 2, \alpha\rangle$ is an OR node: this means that it is sufficient to solve only one of all the immediate successors. On the contrary, $\langle p, q, 1, \lambda\rangle$ is

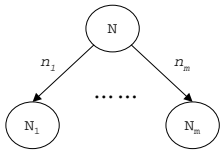


Figure 2: OR node.

an AND node: this means that all the immediate successors must be solved.

A node **N**, labeled $\langle r, s, n, a\rangle$, is *terminal* if one of the following condition holds:

1. $n = 1$ and $\{(r', \alpha) : r \xrightarrow{\alpha} r'\} = \emptyset$
2. $n = 2$ and $\{(s', \alpha) : s \xrightarrow{\alpha} s'\} = \emptyset$

A node fulfilling the condition 1 is said to be *successful terminal*, whereas a node fulfilling the conditions 2 is said to be *unsuccessful* (or *unsolvable*) *terminal*. If condition 1 holds r is not able to move and thus s obviously simulates r. If condition 2 holds q is not able to move and obviously q cannot simulate r, since an action a must be done.

Goal nodes

The goal nodes are all the successful terminal nodes.

Now, in order to apply an heuristic search we have to first formally define an heuristic function $\widehat{h}$ over a node. We define the heuristic only on OR node, i.e. $\langle p, q, 2, \alpha\rangle$, since we want quickly find whether q is able to perform the action α that p has previously performed.

Definition 4.1 ($\widehat{h}(N)$). *Let $N = \langle p, q, 2, \alpha\rangle$ be an OR node. We introduce an auxiliary function $\widehat{h}$ with five arguments, namely $\widehat{h}(p, q, C_q, A_q, S)$, where C_q is a set of constants reachable from q, A_q are set of actions and S is a set of actions. The function $\widehat{h}$ is inductively defined on q as in Table 4. Then, we set: $\widehat{h}(\langle p, q, 2, \alpha\rangle) = \widehat{h}(p, q, \emptyset, \emptyset, \{\alpha\})$.*

The heuristic function guides the construction of the graph towards the searching of two states such that one does not simulate the other. Given an OR node $N = \langle p, q, 2, \alpha\rangle$ the function $\widehat{h}(N)$ associates a non negative value with N, called $\widehat{h}$-value of N. Roughly speaking, that value approximates the number of actions which must be performed to establish that $p \not\preceq q$, that is the minimum number of actions to perform before performing an action α, if any.

$$\mathbf{op}_1 \quad \frac{\{(p',\alpha) : p \xrightarrow{\alpha} p'\} \neq \emptyset = \{(p_1,\alpha_1),\ldots,(p_n,\alpha_n)\}}{\langle p,q,1,\lambda\rangle \overset{\alpha_1}{\rightsquigarrow} \langle p_1,q,2,\alpha_1\rangle \text{ and } \cdots \text{ and } \langle p,q,1,\lambda\rangle \overset{\alpha_n}{\rightsquigarrow} \langle p_n,q,2,\alpha_n\rangle}$$

$$\mathbf{op}_2 \quad \frac{\{q' : q \xrightarrow{\alpha} q'\} \neq \emptyset = \{q_1,\ldots,q_n\}}{\langle p,q,2,\alpha\rangle \overset{\alpha}{\rightsquigarrow} \langle p,q_1,1,\lambda\rangle \text{ or } \cdots \text{ or } \langle p,q,2,\alpha\rangle \overset{\alpha}{\rightsquigarrow} \langle p,q_n,1,\lambda\rangle}$$

Table 3: The operators.

q	$\widehat{h}(p,q,C_q,A_q,S)$
nil	∞
$\beta.q',\ \beta \notin S$ or $\beta \in A_q$	0
$\beta.q',\ \beta \in S$ and $\beta \notin A_q$	$1+\widehat{h}_{aux}(p,q',C_q,A_q,\mathcal{F}irst(p))$
q_1+q_2	$min^*(\widehat{h}(p,q_1,C_q,A_q,S),\widehat{h}(p,q_2,C_q,A_q,S))$
$q_1 \vert q_2$	$\widehat{h}(p,q_1,C_q,A_q,S)+\widehat{h}(p,q_2,C_q,A_q,S)$
$q'\backslash L$	$\widehat{h}(p,q',C_q,A_q\cup(L\cup\overline{L}),S)$
$q'[f]$	$\widehat{h}(p,q',C_q,f^{-1}(A_q),S)$
$y,\ y \overset{\text{def}}{=} q_y,\ y \notin C_q$	$\widehat{h}(p,q_y,C_q\cup\{y\},A_q,S)$
$y,\ y \overset{\text{def}}{=} q_y,\ y \in C_q$	∞

Table 4: Definition of the heuristic function.

The heuristic function $\widehat{h}(p,q,C_q,A_q,S)$ is parametric with respect to a *restriction environment* A_q with $A_q \subseteq \mathcal{V}$, which keeps the set of actions on which some restriction holds on the process q. The function is initially applied to a process with $A_q = \emptyset$. The current environment A_q is modified when the function is applied to $q\backslash L$: in this case the actions in $L \cup \overline{L}$ are added to A_q.

Initially $S = \{\alpha\}$, i.e. the action that q has to perform.

For $q = nil$ the function $\widehat{h}$ returns ∞ as surely q is not able to perform the action α.

When applied to $\beta.q$ the function returns 0 if β is not in S or it is a restricted action (i.e., $\beta \in A_q$). Roughly speaking if β is restricted by A_q then $\beta.q'$ could not be able to move; thus, we optimistically return 0. If $q = \alpha.q'$ and α is not a restricted action, we return 1, as an action α as been found plus the value returned by $\widehat{h}_{aux}$ which is similar to $\widehat{h}$ except that when q is equal to $\beta.q'$, $\beta \in S$ and $\beta \notin A_q$ the function return only 1. The function $\widehat{h}_{aux}$ is applied setting $S = \mathcal{F}irst(p)$ to detect if q is able to simulate the first actions of p.

If q is a choice of two processes q_1+q_2, the minimum number of actions is returned, while is q is a parallel composition the sum is returned. The minimum number is returned by using the following function:

$$min^*(x,y) = \begin{cases} x & \text{if } x \leq y \text{ and } x \neq 0 \text{ and } y \neq 0 \\ y & \text{if } y \leq x \text{ and } x \neq 0 \text{ and } y \neq 0 \\ x & \text{if } y = 0 \\ y & \text{if } x = 0 \end{cases}$$

min^* returns the minimum of two value if both values are different from zero, otherwise either returns zero if both are zero or returns the number different from zero.

If q is of the form $q'\backslash L$, the current environment A_q is modified: in this case the actions in $L \cup \overline{L}$ are added to A_q.

If q is a relabelled process, we must take as set of actions the set $f^{-1}(\rho) = \{\alpha \mid f(\alpha) \in \rho\}$, since now the interesting actions are also those relabelled by f into actions in A_q.

Finally, when q is a constant the function is recursively applied to the body of q. Note that we expand once the body of each constant y, this is obtained storing in C_q each constant which has been expanded. Initially C_q is equal to the empty set.

Note that $\widehat{h}$ is simple to calculate as it is syntactically defined. Once defined the heuristic function that guides the search toward states that are more likely to appear good, we have to defined a goal test function, i.e. a measure or estimate of the distance to reach goal state.

In the following we apply our method on simple example showing encouraging results.

EXAMPLE 4.1. *An example to check simulation is given.*

$$\begin{aligned} p &= a.b.p_1 + c.h.nil \\ q &= a.b.q_1 + c.nil \\ p_1 &= .... \\ q_1 &= .... \end{aligned}$$

The initial node is $N_0 = \langle p, q, 1, \lambda \rangle$*, which is an AND node. Let us apply the operators defined in Table 3. Independently of the definition of* p_1 *and* q_1*, we obtain the following successor nodes:*
$N_1 = \langle b.p_1, q, 2, a \rangle$*, with* $\widehat{h}(N_1) = 2$
$N_2 = \langle h.nil, q, 2, c \rangle$*, with* $\widehat{h}(N_2) = 1$.
According to AO,* N_2 *is chosen for the expansion generating* $N_3 = \langle h.nil, nil, 1, \lambda \rangle$*. It holds that* N_3 *is an unsuccessful node. Thus, we can so conclude that* $p \not\preceq q$ *generating only* 3 *states of* $\mathcal{S}(p)$ *and* 2 *states of* $\mathcal{S}(q)$*. Note that the size of* $\mathcal{S}(p)$ *can be equal to* $4 + |\mathcal{S}(p_1)|$[1]*, while the size of* $\mathcal{S}(q)$ *is equal to* $3 + |\mathcal{S}(q_1)|$.

The following theorem states that the heuristic function, defined in the previous section, is admissible, i.e., it never overestimates the actual cost.

THEOREM 4.1.
Let N *be a node. It holds that*

$$\widehat{h}(N) \leq h(N)$$

where $h(N)$ *is the actual cost of a preferred graph from the initial to a goal node.*
Proof Sketch. *By induction on the structures of the processes* p *and* q *of the node* N.

5. CONCLUSION AND RELATED WORK

A method that uses heuristic searches has been proposed for simulation checking for concurrent systems described in CCS. The novel contributions of our work are the following.

- Application of heuristic search for simulation checking. As far as we know, it is the first attempt to exploit process algebra-based heuristics for equivalence checking in concurrent systems.
- The definition of an admissible heuristic function. In this way, by applying AO*, we can always find the minimal-cost solution graph leading to a node representing two states non equivalent w.r.t a particular equivalence.
- The heuristics is syntactically defined, i.e., it is only based on the CCS specifications of the process, and the proposed method is completely automatic, thus it does not require user intervention and manual efforts.

Recently great interest was shown in combining model checking and heuristics to guide the exploration of the state graph of a system.

In the domain of software validation, the work of Yang and Dill [29] is one of the original ones. They enhance the bug-finding capability of a model checker by using heuristics to search the states that are most likely to lead to an error. In [12] genetic algorithms are used to exploit heuristics for guiding a search in large state spaces towards errors like deadlocks and assertion violations. In [3] heuristics have been used for real-time model checking in UPPAAL. In [1, 22] heuristic search has been combined with on-the-fly techniques, while in [8, 18] with symbolic model checking. Previous works, as for example [7, 16], used heuristic search to accelerate finding errors, while in [26] heuristic search is used to accelerate verification.

As a future work we intend apply this approach also for other equivalences, as for example strong equivalence, weak equivalence and ρ-equivalence, introduced in [2], that formally characterising the notion of "the same behavior with respect to a set ρ of actions": *two transition systems are* ρ**-equivalent** *if a* ρ*-bisimulation relating their initial states exists.*

[1] Given a CCS process p, with $|\mathcal{S}(p)|$ we denote ths size of the transition system of p.

6. REFERENCES

[1] R. Alur, B.-Y. Wang. "Next" Heuristic for On-the-Fly Model Checking. In *Proceedings of the 10th International Conference on Concurrency Theory (CONCUR'99)*, Lecture Notes in Computer Science 1664, 1999. 98-113.

[2] R. Barbuti, N. De Francesco, A. Santone, G. Vaglini. Selective mu-calculus and Formula-Based Abstractions of Transition Systems. *Journal of Computer and System Sciences, 59(3)*, 1999. 537-556.

[3] G. Behrmann, A. Fehnker, T.S. Hune, K. Larsen, P. Petterson, J. Romijn. Efficient Guiding Towards Cost-Optimality in UPPAAL. In *Proceedings of the 7th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, (TACAS'01)*, Lecture Notes in Computer Science 2031, 2001. 174-188.

[4] A. Bouajjani, J.C. Fernandez, N. Halbwachs. Minimal Model Generation. In *Proceedings of the International Conference on Computer-Aided Verification (CAV'90)*, Lecture Notes in Computer Science 531, 1990. 197-203.

[5] E.M. Clarke, O. Grumberg, D.E. Long. Model Checking and Abstraction. *ACM Transactions on Programming Languages and Systems, 16(5)*, 1994. 1512-1542.

[6] E.M. Clarke, O. Grumberg, D. Peled. Model Checking. *MIT press*, 2000.

[7] S. Edelkamp, A. Lluch-Lafuente, S. Leue. Directed Explicit Model Checking with HSF-SPIN. In *Proceedings of the 8th International SPIN Workshop on Model Checking Software*, Lecture Notes in Computer Science 2057, 2001. 57-79.

[8] S. Edelkamp, F.Reffel. OBDDs in Heuristic Search. In *Proceedings of the 22nd Annual German Conference on Artificial Intelligence (KI'98)*, Lecture Notes in Computer Science 1504, 1998. 81-92.

[9] J.-C. Fernandez,A. Kerbrat, L. Mounier. Symbolic Equivalence Checking. In *Proceedings of the 5th International Conference on Computer Aided Verification*, Lecture Notes In Computer Science 697, 1993. 85 - 96.

[10] J.-C. Fernandez, L. Mounier. "On the fly" verification of behavioural equivalences and preorders. In *In Workshop on Computer-aided Verification*, Lecture Notes in Computer Science 575, 1991.

[11] P. Godefroid. Partial-Order Methods for the Verification of Concurrent Systems. Lecture Notes in Computer Science 1032, 1996.

[12] P. Godefroid, S. Khurshid. Exploring Very Large State Spaces Using Genetic Algorithms. In *Proceedings of the 8th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, (TACAS'02)*, Lecture Notes in Computer Science 2280, 2002. 266-280.

[13] P. Godefroid, P. Wolper. Using Partial Orders for Efficient Verification of Deadlock Freedom and Safety Properties. In *Proceedings of the Third International Conference on Computer-Aided Verification (CAV'91)*, Lecture Notes in Computer Science, 575, 1991. 332-342.

[14] J.C. Godskesen, K. Larsen, M. Zeeberg. Tav, tools for automatic verifica tion. In *Proceedings of the 1st International Workshop on Automatic Verification Methods for Finite State Systems* (Grenoble, France), Springer Verlag, 1989.

[15] S. Graf, B. Steffen, G. Luttgen. Compositional Minimization of Finite State Systems Using Interface Specifications. *Formal Aspects of Computing, 8(5)*, 1996. 607-616.

[16] A. Groce, W. Visser. Heuristic Model Checking for Java Programs. In *Proceedings of the 9th International SPIN Workshop(SPIN'02)*, Lecture Notes in Computer Science 2318, 2002. 242-245.

[17] C. Jard, T. Jéron. Bounded-memory Algorithms for Verification on-the-fly. In *Proceedings of the Third International Conference on Computer-Aided Verification (CAV'91)*, Lecture Notes in Computer Science 575, 1991. 192-201.

[18] R.M. Jensen, R.E. Bryant, M.M. Veloso. SetA*: An Efficient BDD-Based Heuristic Search Algorithm. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence (AAAI/IAAI'02)*, AAAI Press, 2002. 668-673.

[19] A. Mahanti, A. Bagchi. AND/OR Graph Heuristic Search Methods. *JACM, 32(1)*, 1985. 28–51.

[20] K. McMillan. Symbolic Model Checking. *Boston: Kluwer Academic Publishers*, 1993.

[21] R. Milner. Communication and Concurrency. *Prentice-Hall*, 1989.

[22] M.O. Moller, R. Alur. Heuristics for Hierarchical Partitioning with Application to Model Checking. In *Proceedings of the 11th IFIP WG 10.5, Advanced Research Working Conference on Correct Hardware Design and Verification Methods (CHARME'01)*, Lecture Notes in Computer Science 2144 , 2001. 71–85.

[23] R. Paige, R. Tarjan. Three partition refinement algorithms. *SIAM J. Com- put.,6 (16),* 1987.

[24] J. Pearl. Heuristics: Intelligent Search Strategies for Computer Problem Solving. *Addison-Wesley.*

[25] D. Peled. All from One, One for All, on Model-Checking Using Representatives. In *Proceedings of the Fifth International Conference on Computer-Aided Verification (CAV'93)*, Lecture Notes in Computer Science 679, 1993. 409–423.

[26] A. Santone. Heuristic Search + Local Model Checking in Selective mu-Calculus. *IEEE Transactions on Software Engineering.* 29 (6) (2003) 510Ű523.

[27] C. Stirling, D. Walker. Local Model Checking in the Modal Mu-Calculus. *Theoretical Computer Science, 89*, 1991. 161-177.

[28] A. Valmari. A Stubborn Attack on State Explosion. In *Proceedings of the Second International Conference on Computer-Aided Verification (CAV'90)*, Lecture Notes in Computer Science 531, 1990. 156–165.

[29] C.H. Yang, D.L. Dill. Validation with Guided Search of the State Space. In *Proceedings of the 35th Conference on Design Automation (DAC'98)*, 1998. 599-604.

Keyword Extraction and Multi-view Clustering Trees for Search and Retrieval in Customer Product Forums

Kivanc Ozonat
HP Labs
1501 Page Mill Road
Palo Alto, CA
kivanc.ozonat@hp.com

ABSTRACT

Customers of large enterprises post threads in company-supported online forums to discuss solutions to product malfunctions, errors and problems. The ability to retrieve the most relevant threads in response to a customer's search query in the product forums requires robust search capabilities. Unfortunately, the lack of (recommendation) links between the threads in the product discussion forums makes it infeasible to use web search algorithms such as PageRank in these forums.

We provide a keyword extraction technique based on term co-occurrences that performs better than traditional tfidf-based techniques. We further provide a search and retrieval algorithm to retrieve the relevant threads in response to a search query in product discussion forums. Our approach is based on a hierarchical, multi-view (thread title and thread content) clustering of the threads. We evaluate our approach and compare with existing statistical algorithms on threads selected from HP's customer support forum.

1. INTRODUCTION

Online product discussion forums consist of *threads*, where each thread typically includes posts by multiple customers discussing a product-related problem. The threads provide useful information to customers who want to find a quick fix for their products, while reducing the workload of the support desks of the manufacturers.

The product forum threads rarely contain links between each other or links from other web sites; thus, it is not feasible to utilize web search algorithms such as PageRank [4, 15] in the product forum settings. Consequently, most forums rely solely on word matching algorithms for search, i.e., the forum search engine retrieves and ranks the threads based on the number of words common to the search query and each thread. Expectedly, this often leads to poor search and retrieval results.

We provide a statistical clustering-based approach to the search and retrieval problem of product threads. Our approach has the following features:

- Term co-occurrence keyword extraction
- Multi-view perspective
- Hierarchical clustering

Keyword extraction is essential to accurate search and retrieval of forum threads. Keyword extraction techniques in the literature are often based on the tfidf method. In the tfidf method, the word frequencies in a repository are compared with the word frequencies in the sample text; if the frequency of a word in the sample text is high while its frequency in the repository is low, the word is extracted as a keyword. In the context of mining customer forums, this approach has a major shortcoming. A customer forum thread typically contains only few sentences and words, making it difficult to obtain reliable statistics based on word frequencies. Many relevant words appear only once in the thread, making it difficult to distinguish them from the other, less relevant words of the thread. We address this issue with term co-occurence keyword extraction, a technique that discovers significant terms in the entire forum and uses the significant terms only to later cluster the similar threads.

During the clustering, we utilize both the thread title and the thread content as features. We note that a thread title (often consisting of just a few words) has a very different characteristic than the thread content (often consisting of at least several sentences), making it challenging to combine the two into one feature vector. To address this issue, we cluster the threads using two views of the data; the title view and the content view.

We cluster the threads in a hierarchical fashion. This way, the customer can be first presented with the threads of the most relevant cluster; if she desires to view more threads, more threads can be included to the presentation by including the threads of the clusters higher in the hierarchy.

The statistical learning literature includes research on the multi-view learning of data [1, 2, 3, 7, 10, 20]. The prior work has focused primarily on multi-view algorithms within a semi-supervised setting. Our setting; however, is unsupervised. In [1], multi-view clustering (with no supervision) has been considered; further, a hierarchical technique has been proposed. The proposed technique builds a single dendrogram by merging the closest clusters based on two distances, one for each of the two views.

We model the training data using Gaussian mixture models, used extensively in literature in the generative statistical clustering of data [6, 8, 10, 11, 12, 14, 15]. The Gaussian mixture models are often designed using the expectation-maximization (EM) algorithm [5]. The EM algorithm assumes that the underlying data follows a Gaussian mixture distribution; consequently, the algorithm assumes that each training sample belongs to each cluster with some membership probability. Unfortunately, the EM updates are intractable in a hierarchical multi-view setting. To address this issue, we use Gauss mixture vector quantization (GMVQ) [8, 11, 12, 13, 16] instead of EM to model the Gaussian mix-

ture.

Designing search strategies for threads in online forums have been discussed in the literature [18, 19], and it has been observed that the PageRank algorithm [4, 15] is not an appropriate choice for retrieving threads since, unlike most web pages, forums do not have the "recommendation" links between each other. In [19], for instance, this issue has been addressed through providing implicit links by constructing a content-based (or more precisely, topic-based) taxonomy of the threads. The assumption is that a surfer is likely to be interested in those threads that have similar topics to the ones she has queried. In [18], a different approach has been taken, and links between threads have been constructed based on the identities of the repliers to the threads. In particular, it is assumed that a replier is more likely to reply to threads in the similar topics than threads in unrelated topics.

Our starting point is similar to the previous work in online forum thread search [18, 19] in that we also argue that links between threads (and the PageRank algorithm based on the links) are not reliable sources when design search techniques for online forums. However, unlike the previous work, we focus on consumer product support forums, which have not been specifically addressed. The product support forums have the characteristic that a customer is interested in only those threads that address her problem, instead of having a desire/interest to jump between related topics. This suggests a clustering-based solution rather than a link-based retrieval. Further, each thread can be viewed as a title-content pair, where the thread can often be summarized with a title and a couple of words. The content, on the other hand, is a relatively long text. We, therefore, base our approach on multi-view clustering of the threads.

Multi-view clustering has been introduced [1, 2, 3, 7, 10, 20] and explored in the semi-supervised learning literature. In [20], a multi-view algorithm has been introduced, where each word in a document has a local view and a global view. The local view is the local context in which a word is used, while the global view is based on the occurrences of the word in other parts of the document. In [2], *co-training* has been introduced, where the training is done using two conditionally independent (given the class labels) views of the training samples. In co-training, first a separate classifier per view is learned using the labeled samples. Then, the highest confidence predictions of each classifier on the unlabeled data are used to construct additional labeled training data.

In [1], multi-view clustering (with no supervision) has been considered; further, a hierarchical technique has been proposed. The proposed technique builds a single dendrogram by merging the closest clusters based on two distances, one for each of the two views. The single dendogram approach proposed in [1], however, is not applicable to our setting; while thread titles (short, often just a few words) are representative of customer queries, thread content (consisting of multiple sentences) are not representative of the queries.

In the statistical clustering literature, techniques can be classified into two categories as generative techniques and discriminative techniques. Most generative techniques have been based on the EM algorithm [5]. The EM algorithm assumes that the training data has been generated by a finite Gaussian mixture distribution, and seeks to find the parameters of that distribution. The EM algorithm has been used extensively within both unsupervised and semi-supervised settings to address problems ranging from document clustering to video classification. We take a different approach, and base our algorithm on GMVQ [8, 11, 12, 13, 16]. The GMVQ has been used both in data compression and image classification. Although there are similarities between GMVQ and EM (for instance, both design a Gauss mixture model based on a training set), the GMVQ is different from EM in two ways: First, in GMVQ, each training sample is assumed to arise from one cluster instead of multiple clusters. Second, the GMVQ does not assume that the underlying data distribution is a Gaussian mixture. GMVQ is a better fit for the hierarchical, multi-view clustering problem than the EM due to the intractability of the EM solution in this setting as discussed in more depth in section 3.

2. PROBLEM FORMULATION

We represent the i^{th} thread in the training set, $1 \leq i \leq N$, by a pair of feature vectors, $x_{i,1}$, the feature vector for the thread title, and $x_{i,2}$, the feature vector for the thread content, where N is the cardinality of the training set. We denote the set for the thread titles by $X_1 = \{x_{1,1}, x_{2,1}, .., x_{N,1}\}$, and the set for the thread content by $X_2 = \{x_{1,2}, x_{2,2}, .., x_{N,2}\}$.

We focus on two clustering decisions; one is a function of the set X_1 and the other is a function of the set X_2. Each of the two functions is designed with guidance from the other with the goal of minimizing the disagreement between the two. Denoting the clustering functions of X_1 and X_2 by $\alpha_1(X_1)$ and $\alpha_2(X_2)$, respectively, the goal is to find the pair of functions α_1 and α_2 that minimizes

$$P(\alpha_1(X_1) \neq \alpha_2(X_2)), \tag{1}$$

where P is an empirical probability.

In order to reduce the effects of overfitting, the minimization in (1) is performed under a constraint on the entropy of clusters. We note that the problem of minimizing (1) with constraints on the entropy of clusters can be viewed a Lagrangian problem with the cost function

$$P(\alpha_1(X_1) \neq \alpha_2(X_2)) + \lambda_v R_v, v = 1, 2 \tag{2}$$

where R_1 is a constraint on the entropy of clusters of α_1, R_2 is a constraint on the entropy of clusters of α_2, and λ_1 and λ_2 are the Lagrangian parameters.

Information-theoretic entropy is frequently used as an overfitting penalty term in designing statistical clustering algorithms. We let R_v, $v = 1, 2$, be the entropy of the clusters, i.e.,

$$R_v = -\sum_{i=1}^{K_v} P(\alpha_v(X_i)) \log P(\alpha_v(X_i)), v = 1, 2, \tag{3}$$

where the probabilities are empirical, and K_v is the number of clusters for α_v.

3. KEYWORD EXTRACTION

Keyword extraction techniques are often based on the tfidf method. The tfidf method compares the word frequencies in the repository with the word frequencies in the sample text; if the frequency of a word in the sample text is high while

its frequency in the repository is low, the word is extracted as a keyword.

While the tfidf-based techniques are known to perform well in many text mining applications, they have a major shortcoming in the context of mining customer forum threads. A customer forum thread typically contains only few sentences and words, making it difficult to obtain reliable statistics based on word frequencies. Many relevant words appear only once in the thread, making it difficult to distinguish them from the other, less relevant words of the thread.

An alternative approach would be to form a vector of keywords in the repository of forum threads, and, generate a binary features vector for each thread. If the i^{th} repository keyword appears in the thread, the i^{th} element of the thread's feature vector is 1, and if the keyword does not appear in the thread, the i^{th} element of the thread's feature vector is 0.

The question with this simple alternative approach is then that of generating keywords in a given repository. We apply three different techniques:

- Remove only stop-words: We only filter out stop words (e.g., if, and, we, etc.) from the repository, and let the vector of keywords be the set of all remaining distinct repository words.
- The tfidf method: We apply the tfidf method on the entire repository by comparing the word frequencies in the repository with word frequencies in the English language. If the frequency of a word is high in the repository and low in the English language, we take the word as a keyword.
- Term co-occurrence: The term co-occurrence method extracts keywords from the repository without comparing the repository frequencies with the English language frequencies. This method is explained next [9].

3.1 Term Co-occurrence

Let N denote the number of all distinct words in the repository of forum threads. We construct a $N \times M$ co-occurrence matrix, where M is pre-selected integer with $M < N$. We typically take M to be 500. We index all distinct words by n, i.e., $1 \leq n \leq N$. We index the most frequently observed M words in the repository by m such that $1 \leq m \leq M$.

The $(n.m)$ element (i.e. ,n^{th} row and the m^{th} column) of the $N \times M$ co-occurrence matrix counts the number of times the word n and the word m occur together. Consider the word ẅirelessẅith index n and the word c̈onnectionẅith index m, and that ẅirelessänd c̈onnectionöccur together 218 times in the repository. Then, the $(n.m)$ element of the co-occurence matrix is 218.

If the word n appears independently from the words $1 \leq m \leq M$ (the frequent words), the number of times the word n co-occurs with the frequent words is similar to the unconditional distribution of occurrence of the frequent words.

On the other hand, if the word n has a semantic relation to a particular set of frequent words, then the co-occurrence of the word n with the frequent words is greater than the unconditional distribution of occurrence the frequent words.

We denote the unconditional probability of a frequent word m as the ex- pected probability p_m and the total number of co-occurrences of the word n and fre- quent terms as c_n. Frequency of co-occurrence of the word n and the word m is written as $freq(n, m)$. The statistical value of χ^2 is defined as

$$\chi^2(n) = \sum_{1 \leq m \leq M} \frac{freq(n,m) - N_n p_m}{n_m p_m}. \quad (4)$$

3.2 Clustering of Frequent Terms

We cluster two or more frequent terms together in either of the following two conditions:

- The frequent words m_1 and m_2 co-occur frequently with each other.
- The frequent words m_1 and m_2 have a similar distribution of co-occurrence with other words.

To quantify the first condition of m_1 and m_2 co-occurring frequently, we use the mutual information between the occurrence probability of m_1 and that of m_2.

To quantify the second condition of m_1 and m_2 having a similar distribution of co-occurrence with other words, we use the Kullback-Leibler divergence between the occurrence probability of m_1 and that of m_2.

4. ALGORITHM

We design a statistical clustering algorithm to minimize (in the Lloyd-optimal sense) the error probability given in (2) under the cluster entropy constraints. Gaussian mixture models have been used extensively in the literature to design generative algorithms for data clustering. Such algorithms are often based on the EM algorithm [5]; however, there also exists alternative training approaches such as Gauss mixture vector quantization (GMVQ) [8, 11, 12, 13, 16]. The EM algorithm assumes that the underlying data follows a Gaussian mixture distribution and tries to fit a Gaussian mixture model to the data, while the GMVQ is a Lloyd clustering algorithm and it does not make any assumptions about the statistics of the underlying data.

We use the GMVQ to design a multi-view hierarchical clustering algorithm. Our choice of GMVQ (over the EM) is motivated by the intractability of the EM solution in minimizing (2) through a hierarchical clustering scheme. We first extend the GMVQ to design a hierarchical (tree-structured) clustering algorithm. We then extend it further to the multi-view setting, where the goal is to minimize the disagreement between two different views of the training data as expressed in (2). The two views in our setting are the thread titles and the thread content.

In section 4.1, we provide an overview of the GMVQ and include the iterative solution for the GMVQ. In section 4.2, we extend the GMVQ to design a tree-structured clustering algorithm. Finally, in section 4.3, we extend it further to the multi-view setting to minimize (2).

4.1 Gauss mixture vector quantization (GMVQ)

Consider the training set $\{z_i, 1 \leq i \leq N\}$ with its (not necessarily Gaussian) underlying distribution f in the form $f(Z) = \sum_k p_k f_k(Z)$. The goal of GMVQ is to find the Gaussian mixture distribution, g, that minimizes the distance between f and g. It has been shown in [8] that the Gaussian mixture distribution g that minimizes this distance

(i.e., minimizes in the Lloyd-optimal sense) can be obtained iteratively with the following two updates at each iteration:

i. Given μ_k, Σ_k and p_k for each cluster k, assign each z_i to the cluster k that minimizes

$$\frac{1}{2}\log(|\Sigma_k|) + \frac{1}{2}(z_i - \mu_k)^T \Sigma_k^{-1}(z_i - \mu_k) - \log p_k, \quad (5)$$

where $|\Sigma_k|$ is the determinant of Σ_k. We note that (5) is also known as the QDA distortion.

ii. Given the cluster assignments, set μ_k, Σ_k and p_k as

$$\mu_k = \frac{1}{\|S_k\|} \sum_{z_i \in S_k} z_i, \quad (6)$$

$$\Sigma_k = \frac{1}{\|S_k\|} \sum_{i} (z_i - \mu_k)(z_i - \mu_k)^T, \quad (7)$$

and

$$p_k = \frac{\|S_k\|}{N}, \quad (8)$$

where S_k is the set of training vectors z_i assigned to cluster k, and $\|S_k\|$ is the cardinality of the set.

4.2 Tree-structured Gauss Mixture Vector Quantization

We use the BFOS algorithm to design a hierarchical (i.e., tree-structured) extension of GMVQ [13]. The BFOS algorithm requires each node of a tree to have two linear functionals such that one of them is monotonically increasing and the other is monotonically decreasing. Toward this end, we view the QDA distortion (5) of any subtree, T, of a tree as a sum of two functionals, u_1 and u_2, such that

$$u_1(T) = \frac{1}{2}\sum_{k \in T} l_k \log(|\Sigma_k|) + \frac{1}{N}\sum_{k \in T}\sum_{z_i \in S_k} \frac{1}{2}(z_i - \mu_k)^T \Sigma_k^{-1}(z_i - \mu_k), \quad (9)$$

and

$$u_2(T) = -\sum_{k \in T} p_k \log p_k \quad (10)$$

where $k \in T$ denotes the set of clusters (i.e., tree leaves) of the subtree T, and μ_k, Σ_k, p_k and the set S_k are as defined in section 4.1.

The magnitude of u_2/u_1 increases at each iteration. This is a key point of the design as, then, pruning is terminated when the magnitude of u_2/u_1 reaches λ, resulting in the subtree minimizing $u_1 + \lambda u_2$.

4.3 Iterative and multi-view TS-GMVQ

We iteratively design two clustering trees, one using the thread title feature vectors, $X_{i,1}$, and the other using the thread content feature vectors, $X_{i,2}$. At each iteration, the two trees are designed (includes tree growing and tree pruning) jointly to minimize (2), the disagreement probability with constraints on the entropy of clusters.

At each iteration, the tree growing starts with a single node tree out of which two child nodes are grown. The Lloyd updates, i.e., (5)-(8), are applied to these two child nodes, minimizing (5), i.e., assigning each training vector to one of the two nodes. Then, one of the two nodes is selected to be split into a pair of new nodes. The selected node is the one, among all the existing nodes, that minimizes (2) after the split. The Lloyd updates, (5)-(8) are then applied to each pair of new nodes, minimizing (10). This procedure of growing a pair of child nodes out of one of the existing nodes, and running the Lloyd updates within the new pair of nodes is repeated until a fully-grown tree is obtained.

We denote the title feature tree by T_1 and the content feature tree by T_2. The trees, T_1 and T_2, are designed using the BFOS algorithm to minimize (2). This implies that, at iteration m, the subtree functionals for T_1 are

$$u_1^m(T) = \sum_{k \in T_1^m} \sum_{x_i \in S_k} P(\alpha_1^m(x_{i,1}) \neq \alpha_2^{m-1}(x_{i,2})), \quad (11)$$

and

$$u_2^m(T) = -\sum_{k \in T_1^m} p_k \log p_k. \quad (12)$$

The u_1 and u_2 functionals for T_2 are analogous.

We observe, by comparing (3) and (12), that

$$\sum_{T_1} u_2^m(T) = R_v \quad (13)$$

and, by comparing (1) and (11), that

$$\sum_{T_1} u_1^m(T) = P(\alpha_1^m(X_1) \neq \alpha_2^{m-1}(X_2)). \quad (14)$$

The u_2^m functional in (12) is identical to the u_2 functional in section 4.2. As for the u_1^m functional, we use (9) for growing the tree and (11) during the pruning. This is possible since (11) is also a linear and monotonically decreasing functional.

Based on the discussion in this section, the multi-view algorithm consists of the following steps (we omit the initialization steps):

i. Grow a TS-GMVQ T_1 tree for the training set $X_{i,1}$, using u_1 and u_2 as given in (9) and (10), respectively.

ii. Grow a TS-GMVQ tree T_2 for the training set $X_{i,2}$, analogously to (i).

iii. Given the tree T_2, prune the fully-grown T_1, using the BFOS algorithm with u_1 and u_2 as given in (11) and (9), respectively.

iv. Given the tree T_1, prune the fully-grown T_2, analogously to (iii).

v. Stop if the change in the cost function, given in (2), from one iteration to the next is less than some ϵ (We set ϵ such that the algorithm stops if the change in (2) is less than 1 percent from one iteration to the next). Else go back to step (i).

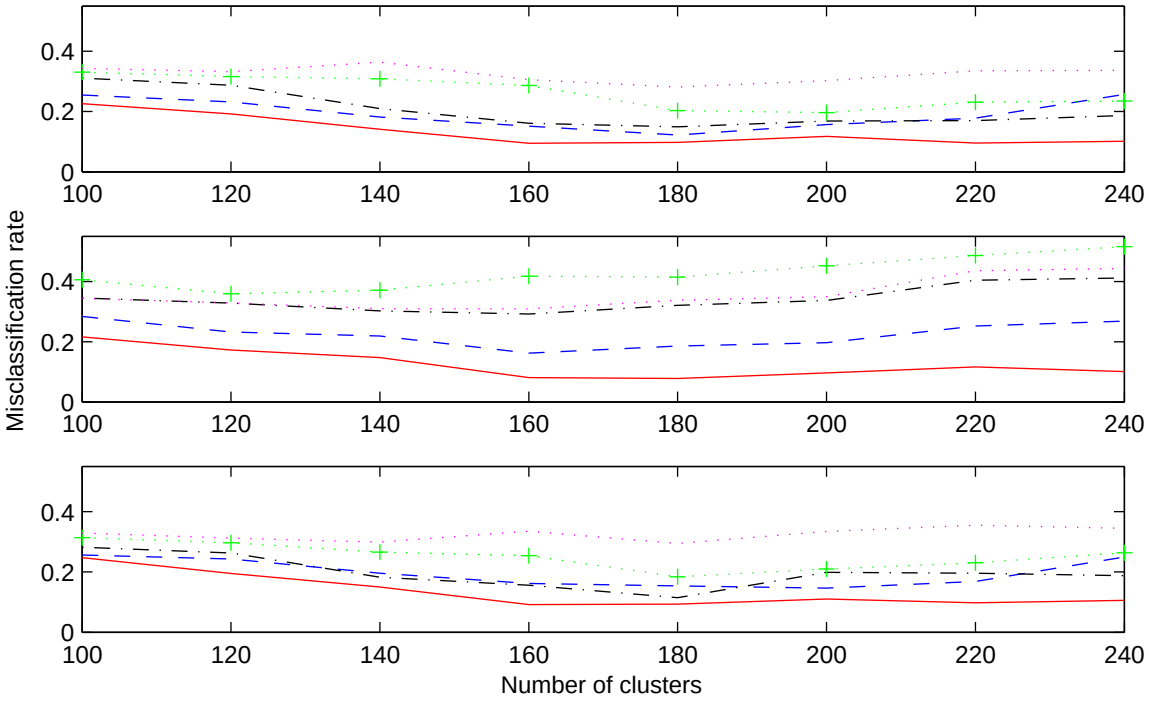


Figure 1: The classification error rate as a function of the number of clusters for the notebook computer thread. The tfidf keyword extraction. The multi-view approach (red solid line) is compared to the GMVQ (blue dashed line), EM (black dash-dot line), TS-GMVQ (magenta dotted line) and agglomerative hierarchical clustering (green dot-plus line). The top plot shows the setting with $y_i = x_{i,1}$, the middle plot shows the setting with $y_i = x_{i,2}$ and the bottom plot shows the setting with $y_i = [x_{i,1}\ x_{i,2}]$.

5. EXPERIMENTS

We've tested the hierarchical multi-view approach on the customer threads included in HP's online support forum. The forum is intended to help the HP customers (both the enterprise customers and the consumers) with troubleshooting their product-related problems. Each thread includes a post by a customer, explaining a problem, followed by replies by others. Each thread has a a title, in addition to thread content.

As of December 2009, the forum included over 2 million postings on issues ranging from operating systems problems for notebook PCs to lockups and freezes for desktops, from mobile web printing problems to problems in enterprise servers and software.

We've compared the hierarchical multi-view clustering approach to 4 other clustering techniques (EM, GMVQ, TS-GMVQ and agglomerative hierarchical clustering) under 3 different feature vector constructions for the notebook PC product line. The comparisons are performed on a subset of 20,000 threads for the product line. In each comparison, we measured the misclassification error, which is the probability of a customer query being assigned to a cluster of a product problem different from the problem expressed in the query. The ground truth for the threads, problems and queries have been established by HP's own customer support teams, independently of the author of this paper.

5.1 Feature vectors

The i^{th} thread has two feature vectors, the thread title $x_{i,1}$ and the thread content $x_{i,2}$. Each feature vector is a W-length binary (0 or 1) vector, where W is the total number of unique words used across all threads. Each word is indexed by w, where $1 \leq w \leq W$.

The w^{th} element of $x_{i,1}$ is 1 if and only if the word w occurs in the title of the i^{th} thread. Similarly, the w^{th} element of $x_{i,2}$ is 1 if and only if the word w occurs in the content of the i^{th} thread. We exclude the stop-words (e.g., and, if, such, etc.) from the feature vectors.

5.2 Comparisons

We've compared the hierarchical multi-view clustering approach to the EM algorithm, standard GMVQ algorithm, TS-GMVQ algorihm and agglomerative clustering. In each comparison, we've tested three different feature vector constructions for the EM, GMVQ, TS-GMVQ and agglomerative clustering: the feature vector is $x_{i,1}$, the feature vector is $x_{i,2}$, and the feature vector is $[x_{i,1}\ x_{i,2}]$.

5.2.1 *Feature vector constructions*

Denoting the feature vector of the training samples by y_i, we consider three feature vector constructions:

i. $y_i = x_{i,1}$. The clustering algorithm takes only the thread titles into account. The thread content are ignored.

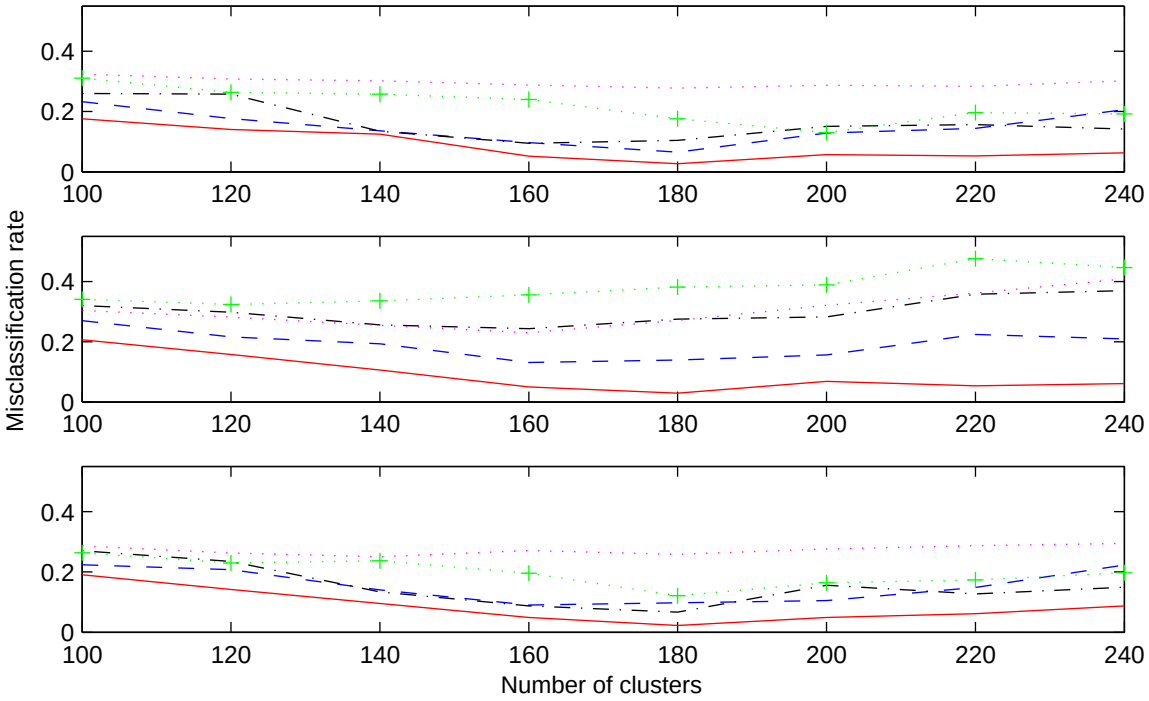


Figure 2: The classification error rate as a function of the number of clusters for the notebook computer thread. The term co-occurrence keyword extraction. The multi-view approach (red solid line) is compared to the GMVQ (blue dashed line), EM (black dash-dot line), TS-GMVQ (magenta dotted line) and agglomerative hierarchical clustering (green dot-plus line). The top plot shows the setting with $y_i = x_{i,1}$, the middle plot shows the setting with $y_i = x_{i,2}$ and the bottom plot shows the setting with $y_i = [x_{i,1}\ x_{i,2}]$.

ii. $y_i = x_{i,2}$. The clustering algorithm takes only the thread content into account. The thread titles are ignored.

iii. $y_i = [x_{i,1}\ x_{i,2}]$. The clustering algorithm takes into account both the thread titles and the thread content.

We compare the hierarchical multi-view approach with the EM, GMVQ, TS-GMVQ and agglomerative clustering under each of these three feature vector construction settings. To avoid any confusion, we emphasize that, the three different feature vector constructions apply only to the EM, GMVQ, TS-GMVQ and agglomerative clustering. The hierarchical multi-view algorithm, on the other hand, always makes use of both $x_{i,1}$ and $x_{i,2}$ in our experiments.

5.2.2 *Compared algorithms*

i. EM: We use the standard EM algorithm. At each iteration, the E-step updates the membership probabilities, $v_{i,k}$, for each cluster k and sample i, while the M-step updates the mean, covariance and the probability of occurrence for each cluster k as

ii. GMVQ: The standard GMVQ algorithm updates are as given in (5)-(8) in section 4.1.

iii. TS-GMVQ: The TS-GMVQ is grown and pruned by the BFOS algorithm using (9) and (10) as described in section 4.2.

iv. Agglomerative hierarchical clustering: We start with N clusters, where N is the number of threads in the training set, and apply agglomerative clustering with the distance criterion given in (5). At each iteration, we merge the two clusters, whose closest feature vectors are closest. In other words, at each iteration, for each pair of clusters, c_k and c_m, we compute the distances between the feature vectors of c_k and the feature vectors of c_m, and denote the closest distance by $d_{k,m}$. We merge the two clusters with the minimum $d_{k,m}$.

5.2.3 *Product line*

We conducted each set of experiments for the threads posted for the notebook PC product line. We used 20,000 threads during clustering, and 10,000 queries for retrieval. For each query, we retrieve the threads of the cluster with its cluster mean μ_k closest to the query under the distance used during clustering. The number of clusters established by the ground truth is 70, i.e., each of the 20,000 threads and 10,000 queries belong to one of the 70 clusters.

5.3 Discussion of Results

Fig. 1 shows the classification error as a function of the number of clusters for the threads in the notebook PC set. The tfidf method is used for extracting keywords. In each plot, the classification error rate of the multi-view approach (red solid line) is compared to that of the GMVQ (blue dashed line), EM (black dash-dot line), TS-GMVQ (ma-

genta dotted line) and agglomerative hierarchical clustering (green dot-plus line). The top plot shows the setting, where the training vector for GMVQ, EM, TS-GMVQ and agglomerative hierarchical clustering are the thread title feature vectors, i.e., $y_i = x_{i,1}$. In this setting, the multi-view approach outperforms the other four algorithms, implying that the inclusion of the thread content to the training process improves the classification.

The middle plot in Fig. 1 shows the setting where the training vectors for the four approaches are the thread content feature vectors, i.e., $y_i = x_{i,2}$. Similar to the setting with $y_i = x_{i,1}$, the multi-view approach leads to the minimum classification error, implying that the thread title should be included in the training to improve the classification accuracy.

The bottom plot in Fig. 1 shows the classification errors for the setting where both the thread titles and thread content are included as feature vectors, i.e., $y_i = [x_{i,1}\ x_{i,2}]$. The performance of the four algorithms (GMVQ, EM, TS-GMVQ and agglomerative hierarchical clustering) all improve compared to the settings shown in the top and middle plots. Once again, this implies that both thread titles and thread content should be included in the feature vectors. However, more importantly, the multi-view approach still outperforms the four algorithms.

It is not surprising for the multi-view approach to outperform the other four approaches in the setting with $y_i = [x_{i,1}\ x_{i,2}]$, because customer queries are more similar to thread titles (short, a few words) than to thread content. Thus, a clustering of thread titles, trained with guidance from the thread content, lead to a more accurate retrieval than a clustering of combined thread titles and thread content.

In Fig. 2, we show the classification error rates. The difference is that here the term co-occurrences are used for extracting the keywords. A comparison with Fig. 1 indicates that, for each clustering strategy, the term co-occurrence keyword extraction leads to a more accurate classification (i.e., lower classification error rates) than the tfidf keyword extraction.

6. CONCLUDING REMARKS

Customers discuss product malfunctions, errors and problems through the threads they post on business-supported online forums. It is not feasible to use web search algorithms such as PageRank for retrieving the relevant threads in response to a search query in the forum since the threads lack recommendation links. We provided a keyword extraction technique based on term co-occurrences that performs better than traditional tfidf-based techniques. Further, we provided statistical, iterative and multi-view approach to searching and retrieving the customer threads in such forums. The hierarhical clustering of the threads makes it possible to present the retrieved threads in a rank-ordered fashion to the customer.

References

[1] S. Bickel and T Scheffer. Multi-view clustering. In Proceedings of the IEEE International Conference on Data Mining, 2004.

[2] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In Proceedings of the Conference on Computational Learning Theory, pages 92-100, 1998.

[3] U. Brefeld and T. Scheffer. Co-EM support vector learning. In Proceedings of the International Conference on Machine Learning, 2004.

[4] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In Proceedings of the 7th International World Wide Web Conference, pp. 107-117, 1998.

[5] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistics Society, 39(1):1-21, 1977.

[6] M. Figueiredo and A. K. Jain. Unsupervised learning of finite mixture models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24:3 , pp. 381-396, March 2002.

[7] R. Ghani. Combining labeled and unlabeled data for multiclass text categorization. In Proceedings of the International Conference on Machine Learning, 2002.

[8] R.M. Gray and T. Linder. Mismatch in high rate entropy constrained vector quantization. IEEE Trans. Inform. Theory, Vol. 49, pp. 1204-1217, May, 2003.

[9] Y Matsuo. Keyword extraction from a single document using word co-occurrence statistical information. International Journal on Artificial Intelligence Tools, Vol. 13, pp. 157-169, 2004.

[10] K. Nigam and R. Ghani. Analyzing the effectiveness and applicability of co-training. In Proceedings of Information and Knowledge Management, 2000.

[11] D.B. O'Brien, M. Gupta, R.M. Gray, and J.K. Hagene. Automatic classification of images from internal optical inspection of gas pipelines. In Proceedings of the ICPIIT VIII Conference 2003, Houston.

[12] D.B. O'Brien, M. Gupta, R. M. Gray, and J. K. Hagene. Analysis and classification of internal pipeline images. In Proceedings of ICIP 2003, Barcelona, Spain.

[13] K. Ozonat, and R.M. Gray. Gauss mixture model-based classification for sensor networks. In Proceedings of the IEEE DCC 2006.

[14] K. Ozonat. Multi-view Clustering Trees for Search and Retrieval in Customer Product Forums. In Proceedings of the ICAI 2010.

[15] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford University, 1998.

[16] K. Pyun, C.S.Won, J. Lim, and R.M. Gray. Texture classification based on multiple Gauss mixture vector quantizers. In Proceedings of Multimedia and Expo, 2002, pp. 501-4, 2002

[17] R. A. Redner and H. F. Walker. Mixture densities, maximum likelihood and the EM algorithm. SIAM Review, 26(2):195-239, 1984.

[18] J. Shen, Y. Zhu, H. Zhang, C. Chen, R. Sun, and F. Xu. A content-based algorithm for blog ranking. Internet Computing in Science and Engineering, 2008.

[19] G. Xu and W.Y. Ma. Building implicit links from content for forum search. In Proceedings of the 29th International Conference on Research and Development in Information Retrieval, 2006.

[20] D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In Proceedings of the 33rd Meeting of the Association for Comp. Linguistics, 1995.

Application of a novel feature selector for human activity recognition based on inertial monitored data

Oresti Baños[1], Miguel Damas[1], Héctor Pomares[1], Ignacio Rojas[1] , Beatriz Prieto[1]

[1] Department of Computer Architecture and Computer Technology, University of Granada, C/Periodista Daniel Saucedo Aranda s/n E-18071, Granada, Spain

Abstract - *The last technological advances in wearable sensors and machine learning are allowing for a new generation of human monitoring techniques, with an especial interest for the analysis of human biomechanics and activity recognition. In this paper, an application of intelligent systems to solve the problem of daily physical activity recognition is presented. Taking into account the importance of data featuring and the selection of the most important features for subsequent pattern recognition stage, a new feature selection methodology based on a filter technique via a couple of two statistical criteria is presented. Satisfactory accuracy rates are achieved by using support vector machines especially for preprocessed data, with remarkable accuracy and applicability in the case of the wrist location.*

Keywords: Activity recognition, Feature selection, Ranking, Knowledge extraction, Support vector machines.

1 Introduction

An important research line is being focused on the monitoring of human biomechanics by using different sensor and methodologies. One of the most important topics in this issue is automatic physical activity recognition, with an increasingly outcome in healthcare related applications, particularly centered on the aged people [15]. Supporting a more independent life and mobility for the elderly, improving their welfare and health status and facilitating more autonomous activities regardless the location is one the most important objectives of this kind of applications.

Vision [9, 12], microphones [13] or ambient sensors [6] are widely used, but attending for the simplicity and the capabilities of the new generation of mobile systems, inertial sensors have been considered as the most interesting approach. In this work we present a particular methodology stressing on feature selection stage to extract a set of the most important features to be used in activity recognition, considered extremely important to ensure the performance of the subsequent pattern recognition system employed. One of the most important characteristic of the method proposed is that we do not provide a rank order for every individual feature but for every set of features, allowing for the synergic utility of several features when considered together at the same time.

The rest of the paper is organized as follows: In Section 2 a brief summary of the experimental setup and the data preprocessing is made. Next, the rank-based feature-set selection methodology developed is presented, describing the fundamentals of this method and the algorithm's main steps. Section 4 presents the results obtained for the performance of the method for a specific example, comparing the accuracy results with related previous works.

2 Data processing and featuring

The initial setup starts from a signal set corresponding to acceleration values measured by a group of sensors located in strategic different parts of the body (hip, wrist, arm, ankle, thigh), for four daily activities (walking, sitting, standing, running) [1].

Distortion elements related to system monitoring and processing, along with the random character of the individual execution determine that a particular data processing and analysis should be done. One goal of this work is to evaluate the importance of this preprocessing stage. It is well-known that filtering techniques normally entail loss information, so it would be interesting to compare the results in contrast to work directly with the original raw signals. According to this, two processing techniques are respectively tested, consisting of a mean filter (Fig. 1.b) and a band pass filter (Fig. 1.c). The mean filtering is defined to remove the initial offset introduced in the original data acquired and the discontinuities associated to the sensors calibration changes between different monitoring sessions (Fig. 1.a). Band pass filtering also permits to remove the high frequency noise. Considering that a 20 Hz sampling is sufficient to assess habitual daily physical activity [8], an elliptic filter with 0.5Hz and 20 Hz cutoff frequencies is used for the last one.

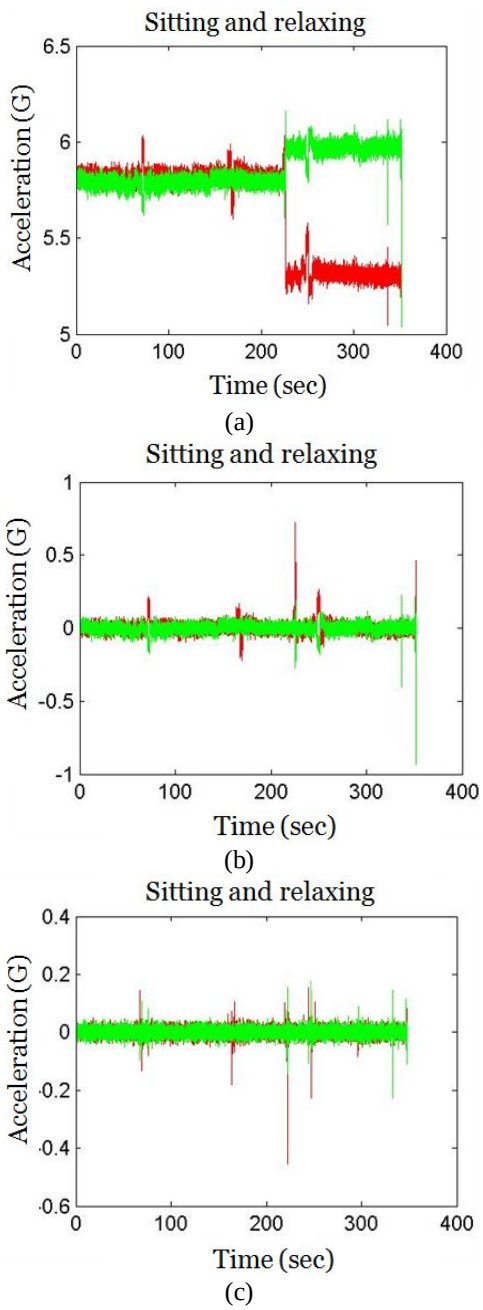


Figure 1. Signals (green = x-axis, red = y-axis) corresponding to the activity 'sitting and relaxing', monitored through the wrist accelerometer. a) Original data with a 5.8G offset and a discontinuity at second 220 approx., b) Mean filtered data and c) BP filtered data.

Once data have been processed, a parameter set made up of 861 features corresponding to a combination of statistical functions such as mean, kurtosis, mode, variance, etc., and magnitudes obtained from a domain transformation of the original data such as energy spectral density, spectral coherence or wavelet decomposition, among others. These features are evaluated over the complete signal, although other alternatives based on windowing and sub-segmentation signal feature extraction could also been tested.

In this stage, we now must rely on a feature selection process that has the responsibility of deciding which features or magnitudes are the most important ones to decide the kind of activity the person is carrying out. In the next section we describe the method we have designed to accomplish this task.

3 Feature selection based on discrimination and robustness statistical criteria

To obtain a specific group of variables from a big initial set is not a trivial task because of the huge number of possible feature combinations. In our experimental setup the sample space is represented by n = 861 features, so brute force techniques like 'branch and bound' (O(2n) convergence 2861 ≈ 1.5 x 10259 possible permutations) or wrapper methods are impractical. In this section, we present an alternative method based on the concepts of discrimination and robustness for a complete set of features.

Let us define the sample range of a class as the set of values included between the maximum and the minimum value (both inclusive) that a feature or variable takes for this class. Given a group of samples (associated to every class) we rank its discriminant capability with respect to that class through the overlapping probability between this class and the others. This is calculated computing the number of samples from the analyzed class which are inside of the sample range defined by the others. For N classes and M samples for each class (let us suppose that this number is independent of the class), we define the overlapping probability of a set of samples as follows:

$$p(k) = \frac{1}{N-1}\sum_{n\neq k}^{N}\frac{m(k,n)}{M} \qquad (1)$$

with *m(k,n)* being the number of samples from the class *k* inside the sample range of class *n*.

We now carry out a thresholding process which allows us to define the feature analyzed as discriminative or not. This *overlapping* threshold takes values from 0 (the most restrictive, for cases with no overlapping between classes) to 1 (the most relaxed, when every sample from a class is

inside the others). In general for a specific feature, if the analyzed class exceeds the threshold, the feature will be considered as no discriminant for this class.

Table 1. Example for 4 classes and 5 sources of quality feature set (ranking) based on discriminant (number of activities discriminated, first column) and robustness (number of motion sensor where the feature is discriminant, second column) criterions.

Discriminant capacity	Robustness	Quality group
4	5	#1
	4	#2
	3	#3
	2	#4
	1	#5
3	5	#6
	4	#7
	3	#8
	2	#9
	1	#10
2	5	#11
	4	#12
	3	#13
	2	#14
	1	#15
1	5	#16
	4	#17
	3	#18
	2	#19
	1	#20
0	-	#21

Apart from the discriminant capacity of a feature or a set of features, a second characteristic is now defined which takes into account the usability of this set of features in different information contexts or sources. For instance, a specific measure taken from the hip accelerometer can be very discriminative to distinguish between the activities *walking* and *standing still*, but this very same measure may not be that reliable when taken from the ankle accelerometer.

There may be some measures with the same discriminant capability between those activities which are not so dependent of the exact location of the sensor or, at least, which are still reliable when taken from a bigger number of sensors. We will denote this measure as the robustness criterion of a set of features. In short, discriminant capacity says how useful a motion feature is in general, and robustness is how this depends on where the sensor is.

Combining both criteria we obtain a quality ranking procedure capable of grouping features in different stages. For the sake of simplicity, let us suppose a recognition system with 4 classes and 5 sources; features will be classified in groups defining a ranking (see table 1). For instance, features that discriminate 4 classes in every source will be added to group #1 (the best). Group #13 will be completed with features that classify 2 classes (the same) in 3 sources at least. This example is extensible to any classes and sources.

4 Results

Most remarkable features (set #1 and #2 primarily) are geometric mean for amplitude signal, autocorrelation and some wavelets coefficients obtained through a 3-level Daubechies decomposition. For classification paradigm, taking into account the good results obtained in several machine learning previous works, and even not being much used in activity recognition studies related, the knowledge inference system is defined through support vector machines (SVM) [11], using a RBF kernel implementation with hyper-parameters γ and C automatically tuned using a grid search. A 10-fold cross validation method is used for training and testing. Results are showed in Figure 2.

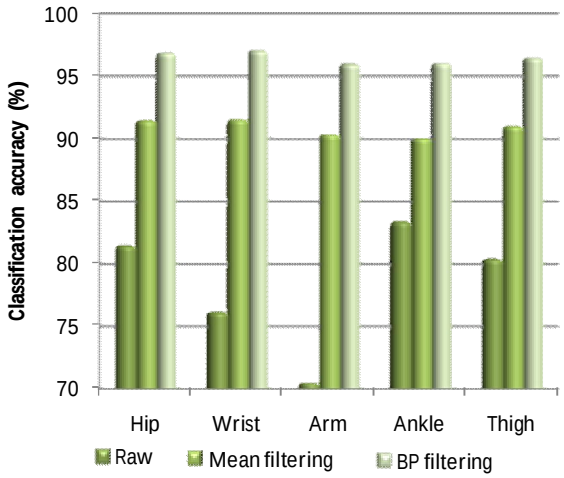


Figure 2. Accuracy rates for the three processing approaches. Results are identified with the corresponding sensor label depending on the location of the source data.

Clearly, the preprocessing is needed to optimize the recognition capability of the system, with particular remarkable results for the band pass (BP) approach. In fact, the mean filtering, that could be reasonably interpreted as a low pass filter (to remove the DC bias or 0Hz component) determines an important improvement from the unprocessed data. Notwithstanding, the BP filter also permits the removal of the frequency components above 20Hz. This demonstrates that there are irregularities and abnormalities in upper frequencies that make more difficult the discrimination task, at least for the four activities analyzed.

Besides good results are obtained in general for the BP filtering approach, we want to stress on the importance that these results are achieved for each sensor separately, it means, no information from other sensors is needed to

accomplish the activity recognition task. Considering that accuracy rates above 95% for all the sensors any studied placement will work efficiently for the activities analyzed, but particularly interesting is considered the wrist sensor location because of the highest experimental accuracy (>97%) and especially for its unobtrusive properties.
Although a strict comparison with other studies cannot be made since the data and the number of classes may differ, in [7] a 83-90% classification accuracy was reached, 92.85%-95.91% in [5], 89% in [1], or 93% and 89% on recent works ([2] and [3] respectively).

5 Conclusions

In this work we have very briefly shown a direct application of ranking selection methods used on daily physical activity automatic recognition. An efficient classification method requires a productive and limited feature set, being necessary a selection process since the initial set is quite huge. We have defined a feature selector based on statistical discrimination and robustness criteria, focused on low computational time and resources, defining a real alternative to other selection processes.

The importance of an adequate preprocessing stage has been also showed, demonstrating that singularities and irregularities affect physical activity monitored data. The wrist location stands out for its efficiency and unobtrusively.

For future work, we aim to make a time-based comparison to traditional features selectors [4,10,14].

6 Acknowledgments

This work was supported in part by the Spanish CICYT Project TIN2007-60587, Junta de Andalucia Projects P07-TIC-02768 and P07-TIC-02906, the CENIT project AmIVital, of the "Centro para el Desarrollo Tecnológico Industrial" (CDTI- Spain), the FPU Spanish grant AP2009-2244 and the UGR grant "Iniciación a la Investigación 2010/2011". We want to express our gratitude to Prof. Stephen S. Intille, Technology Director of the House_n Consortium in the MIT Department of Architecture for the experimental data provided.

7 References

[1] Bao, L., Intille, S.S.: Activity Recognition from User-Annotated Acceleration Data. In: Activity Recognition (Pervasive Computing) ». LNCS, vol. 3001, pp. 1--17. Springer, Heidelberg. (2004)

[2] Bonomi, A.G., Goris, A.H.C., Yin, B., Westerterp, K. R.: “Detection of Type, Duration, and Intensity of Physical Activity Using an Accelerometer”. Medicine & Science in Sports & Exercise. vol. 41, pp. 1770--1777. (2009)

[3] Ermes, M., Pärkka, J., Mantyjarvi, J., Korhonen, I.: “Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions”. IEEE Trans. Inf. Technol. Biomed. vol. 12, pp. 20--26. (2008)

[4] Kohavi, R., Sommereld, D.: “Feature Subset Selection Using the Wrapper Method: Overtting and Dynamic Search Space Topology”. First International Conference on Knowledge Discovery and Data Mining (1995)

[5] Lee, S.W., Mase, K.: Activity and location recognition using wearable sensors. IEEE Pervasive Computing. vol. 1, pp. 24--32. (2002)

[6] Logan, B., Healey, J., Philipose, M., Tapia, E.M., Intille, S. S.: “A long-term evaluation of sensing modalities for activity recognition”. In: Proceedings of the 9th international conference on Ubiquitous computing, pp. 483--500 (2007)

[7] Mantyjarvi, J., Himberg, J., Seppanen, T.: “Recognizing human motion with multiple acceleration sensors”. In: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, pp. 747--752. (2001)

[8] Mathie, M.J., Coster, A.C.F., Lovell, N.H., Celler, B.G.: “Accelerometry: providing an integrated, practical method for long-term, ambulatory monitoring of human movement”. In: Physiol. Meas., vol. 25, pp. 1--20. (2004)

[9] Ryoo, M.S., Aggarwal, J.K., Patras, I.: “Semantic representation and recognition of continued and recursive human activities”. In: International Journal of Computer Vision, vol. 82, pp. 1--24. (2009)

[10] Song, L., Smola, A., Gretton, A., Borgwardt, K.M., Bedo, J.: “Supervised feature selection via dependence estimation”. In: Proceedings of the 24th international conference on Machine learning, pp. 823--830. (2007)

[11] Vapnik, V. “Statistical Learning Theory”, John Wiley & Sons. (1998)

[12] Wang, Z., Li, B.: “Human activity encoding and recognition using low-level visual features”. In: Proceedings of the 21th International Joint Conference on Artificial Intelligence, pp. 1876--1882. (2009)

[13] Ward, J., Lukowicz, P., Troster, G., Starner, T.: "Activity Recognition of Assembly Tasks Using Body-Worn Microphones and Accelerometers. Pattern Analysis and Machine Intelligence », IEEE Transactions on, vol. 28, pp. 1553--1567. (2006)

[14] Xu, Z., Jin, R., Ye, J., Lyu, M.R., King, I.: "Non-monotonic feature selection". In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 1145--1152. (2009)

[15] Yu, X.: "Approaches and principles of fall detection for elderly and patient. In: e-health Networking", Applications and Services, 2008. HealthCom 2008. 10th International Conference on, pp. 42--47. (2008)

Improvement of Personalized Recommendation Algorithm Based Content-boosted Collaborative Filtering Algorithm

B. Raja Sarath Kumar[1], B. John Ratnam[2] and M. S. Prasad Babu[3]

[1]CSE Department, Lenora College of Engineering, Rampachodavaram, E. G. District, India
[2]BS&H Department, Lenora College of Engineering, Rampachodavaram, E. G. District, India
[3]CS&SE Department, Andhra University College of Engineering, Visakhapatnam, India

Abstract

Collaborative filtering (CF) systems have been proven to be very effective for personalized and accurate recommendations. These systems are based on the Recommendations of previous ratings by various users and products. Since the present database is very sparse, the missing values are considered first and based on that, a complete prediction dataset are made. In this paper, some standard computational techniques are applied within the framework of Content-boosted collaborative filtering with imputational rating data to evaluate and produce CF predictions. The Content-boosted collaborative filtering algorithm uses either naive Bayes or means imputation, depending on the sparsity of the original CF rating dataset. Results are presented and shown that this approach performs better than a traditional content-based predictor and collaborative filters.

Key Words: collaborative filtering, Mean Absolute Error, Significance Weighting Factor, Harmonic Mean weighting factor

1. Introduction

World Wide Web has created the universe as global village, with an explosive growth of enormous information. Getting the relevant information from the internet is a very big problem. Personalized recommendation systems may be used to get relevant information from the internet. CF is a widely used technique for information filtering in personalized recommendation systems. They are mainly classified into three categories: namely, memory based collaborative filtering, model based collaborative filtering and hybrid or content boosted collaborative filtering techniques [5]. Memory based CF is further sub-divided into user-based collaborative filtering, item based collaborative filtering and content based collaborative filtering techniques. User-based collaborative filtering recommendation systems depend on item rating predictions. The process of considering items to a user is based upon the opinions of people with similar likes or dislikes. Generally the filtering system determines which users have similar likes or dislikes. Recommender systems helps to users to overcome information overload by providing personalized suggestions based on a history of a user's likes and dislikes. On-line web portals like Amazon.com [16]; eBay.com [17]; Netflix.com [18] etc. are providing these recommending services to know about the personalized interests.

Collaborative Filtering systems recommend an item based on ratings in a given domain by duly examining the likes or dislikes among the other users likes or dislikes. On the other hand, content-based methods in Collaborative Filtering provide recommendations by comparing representations of content contained in an item to representations of content that interests the users. Content-based methods can concatenate each user, whereas it is not possible in Collaborative filtering but it still has some other key advantages. Collaborative filtering can also perform with less data of items to recommend items that are relevant to the user, but do not contain content. The main draw backs of these algorithms are sparsity [5] and cold-start [2] problems. Sparsity occurs when the user do not rate more items. In this case the sparse user-item rating value decreases which causes low value ratings in finding similar set of users. Cold-start is a basic problem concerned with an item that can only be recommended after rated by a user [9].

To resolve these drawbacks Content-boosted Collaborative Filtering Algorithm is considered by converting content-based predictions into a sparse user rating matrix with full rating matrix. This approach performs significantly and produces better results than User-based or Item-based or Content-based Collaborative Filtering algorithms. This paper puts forward a comprehensive study of the Content boosted collaborative filtering algorithm and its implementations.

2. Collaborative Filtering

Content Based Collaborative Filtering first proposed by [9] uses neighborhood-based

recommendations. Here subsets of users are chosen based on their similarity to the active user, and a weighted combination of their ratings is used to produce predictions for the active user. The algorithm is given in the following steps.

Step1. All active users are equally weighted with respect to similarity.

Step2. Measure the similarity between users as the Pearson Correlation Coefficient between their ratings vectors. Pearson correlation coefficient can be computed by the formula.

$$P_{a,u} = \frac{\sum_{i=1}^{rn}(r_{a,i} - \bar{r}_a) \times (r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i=1}^{rn}(r_{a,i} - \bar{r}_a)^3 \times \sum_{i=1}^{rn}(r_{u,i} - \bar{r}_u)^3}} . \quad (1)$$

Where

$\mathbf{r}_{a,i}$ is the rating given to item i by user a

r_a is the mean rating given by user a.

Step3. Select the n users which are having the highest similarity with the active user.

Step4. Form the neighborhood of active user recommendations predictions are computed as the weighted average of deviations from the neighbor's mean:

$$P_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^{n}(r_{u,i} - \bar{r}_u) \times P_{a,u}}{\sum_{u=1}^{n} P_{a,u}} \quad .. (2)$$

Where

$P_{a,i}$ is the prediction for the active user a for item i.

$P_{a,u}$ is the similarity between users a and u.

n is the number of users in the neighborhood.

Step5. Compute the prediction of a weighted combination of the neighbors' ratings.

2.1 Significance Weighting Factor (SWF)

Significance weighting factor is used to devalue the correlations generated based on few co-rated items to prevent bad predictors. If the number of co-rated items (n) is less than 50 then SWF is the product of their correlations. When n ≥ 50 then the factor $Sg_{a,u} = 1$.

3. Content based Predictor Algorithm

The implementation process starts with a bag-of-words naive Bayesian text classifier to learn a user profile from a set of rated movies. Prediction task can be assumed as a text-categorization problem and Movie content information can be assumed as text documents, and user ratings (1-5) can be as one of six class labels. Multinomial text model is adopted [5], in which a document is modeled as an ordered sequence of word events drawn from the same vocabulary, V. The naive Bayes assumption states that the probability of each word event is dependent on the document class. For each class C_i , and word, $w_k \in V$ V, the probabilities, $P(C_i)$ and $P(w_k \mid C_i)$ can be evaluated from the training data. The subsequent probability of each class given a document D, is computed using Bayes rule.

$$P(C_i \mid D) = \frac{P(C_i)}{P(D)} \prod_{i=1}^{|D|} P(a_i \mid C_i) \quad \ldots(3)$$

where a_i is the ith word in the document,

and | D |is the number of words in the document.

In the implementation process, movies are represented as a vector of documents", d_m, one for each slot, the probability of each word given the category and the slot, $P(w_k \mid C_{i,} S_m)$must be estimated. The subsequent category probabilities for a film, F, computed using:

$$P(C_i \mid F) = \frac{P(C_i)}{P(F)} \prod_{m=1}^{S} \prod_{i=1}^{|d_m||} P(a_{mi} \mid C_{i,} S_m) \ldots(4)$$

where S is the number of slots

and a_{mi} is the i[th] word in the m[th] slot.

3.1 Algorithm

Each example in *Examples* is a vector of bag-of-words and a category corresponding to a 0-5 rating. Each bag of bag-of-words corresponds to a slot e.g. title, cast, reviews, etc. C is the set of all possible categories. This function estimates the probability terms $P(a_{mi} \mid C_{i,} S_m)$ describing the probability that a randomly drawn word from a slot S_{rn} in an example in class C_i will be the word a_{mi}.

STEP 1.

$Calculatec \quad lassprior \quad , (P(C_i)$

$docs_j \leftarrow subsetofdocumentsfromExamples$

$forwhichtheclasslabelis \quad j$

$$P(C_j) \leftarrow \frac{|docs_j) + \frac{1}{|Examples|}}{|Examples| + \frac{|c|}{|Examples|}}$$

STEP 2.

$Calculateconditionalprobabilites, P(a_{mi} \mid C_{j,} S_m$

$foreachslot\ S_m$

$Vocabulary_m \leftarrow$ set of all district tokens occurring in slot S_m in all examples.

For each possible class C_j

- $Text_{mj} \leftarrow$ a single document created by concatenating all bags – of – words appearing in slot S_m and in class C_j
- $n \leftarrow$ total number of district word positions in $Text_{mj}$
- For each token, a_{mi} in $Vocabulary_m$
  * $n_k \leftarrow$ number of times token a_{mi} occurs in $Text_{mj}$

$$P(a_{mi} \mid C_i, S_m) \leftarrow \frac{n_k + \frac{1}{|Examples|}}{n + \frac{|Vocabulary_m|}{|Examples|}}$$

4. Content boosted Collaborative Filtering Algorithm

In Content boosted collaborative filtering, pseudo user-rating vectors of all users are combined into pseudo rating matrix. Similarity between the active users is computed by using the Pearson correlation coefficient. the following steps describe the proposed algorithm

Step1. A pseudo user-rating vector for all users in the database is created by Using Harmonic Mean Weighting Factor (HMW). The pseudo user-ratings vector, v_u, consists of the item ratings provided by the user u. $r_{u,i}$ denotes the actual rating provided by user u for item i, while $C_{u,i}$ is the rating predicted by the content-based system.

$$\mathbf{V_{u,i}} = \begin{cases} r_{u,i} : if user\ u\ rated item\ i \\ \mathbf{C_{u,i}} : Otherwise \end{cases} \quad \ldots\ldots (5)$$

Step2. Compute pseudo rating matrix V by combine the pseudo user-ratings vectors of all users.

Step3. Compute the similarity between active user a and another user u using the Pearson correlation coefficient by the pseudo user-ratings vectors v_a and v_a.

Step4. Computed mean-centered ratings of the best-n neighbors of that user as weighted sum of the active user by incorporate a Self Weighting factor in the final predictions.

Step5. Combined the above two weighting schemes to evaluate the final CBCF predictions.

4.1 Harmonic Mean Weighting

An accuracy of a pseudo user-ratings vector is computed for a user depends on the number of movies rated. If the user rated many items, the content-based predictions are good and hence his pseudo user-ratings vector is fairly accurate. Otherwise, if the user rated only a few items, the pseudo user-ratings vector will not be as accurate. It is clear that inaccuracies in pseudo user-ratings vector often yielded misleadingly high correlations between the active user and other users. Harmonic Mean weighting factor (HMW) is used to incorporate these low user-rated correlations.

$$hm_{i,j} = \frac{2m_i m_j}{m_i + m_j}$$

$$\mathbf{m_i} = \begin{cases} (n_i/50) : if n_i < 50 \\ \mathbf{1} : Otherwise \end{cases} \quad \ldots. (6)$$

In the above equation, n_i refers to the number of items that user i has rated. The harmonic mean tends to bias the weight towards the lower of the two values namely m_i and m_j .Thus correlations between pseudo user-ratings with at least 50 user-rated items each, will receive the highest weight, regardless of the actual number of movies each user rated. Otherwise, even if one of the pseudo user-rating vectors is based on less than 50 user-rated items, the correlation will be devalued appropriately. The threshold 50 is based on the learning curve of the content predictor. It can be noted from [5] that initially as the predictor is given more and more training examples the prediction performance improves, but at around 50 it begins to level off. Beyond this is the point of diminishing returns; as no matter how large the training set is prediction accuracy improves only marginally. The HMW includes the significance weighting to obtain the hybrid correlation weight.

$$hw_{a,u} = hm_{a,u} + Sg_{a,u} \ldots. (7)$$

4.2 Self-Weighting

The mean-centered votes of the best-n neighbors of that user are computed as weighted sum of the active user. In our approach, pseudo active user is also added to the neighborhood. The other neighbors are given more importance than pseudo active user. A Self Weighting factor has been incorporated in the final prediction:

$$\mathbf{SW_a} = \begin{cases} (n_a/50) * \max : if n_a < 50 \\ \max : Otherwise \end{cases} \quad \ldots. (8)$$

where

n_a is the number of items rated by the active user.

Combining the above two weighting schemes, the final CBCF prediction for the active user a and item i is produced as follows:

$$p_{a,i} = \bar{v}_a + \frac{sw_a(c_{a,i} - \bar{v}_a) + \sum_{\substack{u=1 \\ u \neq a}}^{n} hw_{a,u} p_{a,u} (v_{u,i} - \bar{v}_u}{sw_a + \sum_{\substack{u=1 \\ u \neq a}}^{n} hw_{a,u} P_{a,u}} \quad \ldots. (9)$$

Where

$C_{a,i}$ Corresponds to the Content predictions for the active user and item i.

$v_{y,i}$ is the pseudo user-rating for a user u and item i

$\bar{v}_u$ is the mean over all items for that user.

Sw_a, $hw_{a,u}$, and $P_{a,u}$ are evaluated.

n is the size of neighborhood.

5. Implementation and improvement of the CBCF Algorithm

5.1. Methodology

The experimental methodology used for computing the different prediction algorithms namely CF predictor, Content-based algorithm and CBCF are presented and evaluated. All the experiments were performed on an Intel Pentium-IV Processor, 2 GB RAM system and implemented in Java and executed. The dataset is stored in database MySQL in the same computer.

5.2. Dataset

To carry out the research and analysis for content boosted collaborative filtering system, the GroupLens Research Project agency at the University of Minnesota developed Internet Movie Database which contains the user-movie ratings and the movie details, called the **EachMovie dataset** [12]. This dataset may be used to derive the results. The data set consists of 100,000 ratings (1-5) from 943 users on 1682 movies. Each user has rated at least 20 movies. It provides demographic data such as age, gender, and the zip code supplied by each person. The content of the information of every movie is considered as a set of slots. Each slot is represented by number of words. Further, the data has been segregated and discarded for having less than 20 ratings or in complete demographic information. Now the dataset provides the actual rating data provided by each user for various movies and used in the implementation process to generate prediction values for various algorithm systems. The same was downloaded and has been used from their website [13] to derive the results.

5.3. Experimental Evaluation

A subset of the ratings data from the EachMovie data set used for the purposes of comparison. 20% of the users were randomly selected to be the test users. In the each movie data downloaded from grouplens website [13], it mentioned that the data sets u1.base and u1.test through u5.base and u5.test are 80%/20% splits of the u data into training and test data. Each of u1, u2, u3, u4, and u5 has disjointed test sets for cross validation. These data sets can be generated from u.data by mku.sh. Source file u.data contained the u dataset by 943 users with 100000 ratings on 1682 items. Each user has rating at least 20 movies. This is a tab separated list of user id, item id, rating and timestamp.

The main java classes designed and developed to evaluated the predictions for the content-based algorithm and content-boosted algorithm are *CBA5.java, NBSSimblanceRow.java, Probability.java XYSplineRendererDemoTest.java* . A segment of java code snippet and the structure of the java classes that implements the content-based collaborative filtering, collaborative filtering predictor and Content-Boosted Collaborative Filtering algorithms proposed in the system is as follows.

```
List original = new ArrayList ();
String fileName2 = "D:\\Excelwork\\ml-data_0\\u.data";
int usersSize = 100;
int itemsSize = 1000;
we.initialize(original, usersSize, itemsSize);
we.populateFileToList(original, fileName2, usersSize, itemsSize);
```

Here the List '*original*' is the list which contains the original ratings of the users which will be compared with the predicted ratings. It is designed to populate the list with the ratings read from the u.data file with the mentioned Path in the code.

```
List test = new ArrayList();
fileName2 = "D:\\Excelwork\\ml-data_0\\u5.test";
we.initialize(test, usersSize, itemsSize);
we.populateFileToList(test, fileName2, usersSize, itemsSize);
```

The List '*test*' is the list which contains the test ratings of the users. Test data is the subset of original data. Using test data, it is designed to produce the user rating predictions and populating the 'test' list with values read from u5.test. Content based predictions are generated by treating the task as a text-categorization problem. The movie data which contains the content

information is considered as text documents, and user ratings given as 1-5.

```
fileName2 = "D:\\Excelwork\\ml-data_0\\u.genre";
ArrayList genre = new ArrayList();
we.initializeGenre(genre, fileName2);
```

The List '*genre*' is the list of genre of the movies. Each movie genre is given a unique number which is used in item classification.

```
fileName2 = "D:\\Excelwork\\ml-data_0\\u.item";
List items = new ArrayList();
we.initializeItems(items, 1682, 30);
we.populateItemsToList(items, fileName2, 1682, 30, genre);
```

The List 'items' is the list of all the items that presented in u.item. 1682 is number of items given, and 30 is the number of properties mentioned in the u.item file. It is designed and developed to populate the test list with values read from u.item. All the properties are embedded in a child list and the child list is added to parent list.

```
List docsIJ = new ArrayList();
we.initialize(docsIJ, usersSize, 5);
we.populateNoOfRatingsVsClazz(docsIJ, test, usersSize);
```

The List docsIJ is the list that contains the data of users which rated for first grade(rating given as 1). Similarly second grade and so on ratings given by particular user. Hence it contains number particular ratings (1-5) which was graded by each and every user. The method populateNoOfRatingsVsClazz is designed to develop the list of the ratings of all users.

```
List Examples = new ArrayList();
we.populateExamples(Examples, docsIJ);
```

'Examples' is the list of number of total ratings given by every user. It is designed and developed to generate MAE values for content boosted collaborative filleting. MAE calculates the irrelevance between the recommendation value predicted by the system and the actual evaluation value rated by the user. The measurement method of evaluating the recommendation quality of recommendation system mainly includes statistical precision measurement method it includes to measure the recommendation quality [6]

The generated prediction values are stored in an arraylist of Examples mentioned above and tabulated in the next sections. This arraylist Example is input for the the class *populatePredictContentBoostedUJ* to generate the MAE values. The arraylist contains the values of the predicted user rating is geneated with the java class 'Examples' and actual user rating arraylist generated with the java class 'original'. These two arraylist are the inputs for *populatePredictContentBoostedUJ java class to generate MAE values*

```
sheetData1 = ((List) ((ArrayList) test).clone());
    s3 =new ArrayList();
   Hashtable table = new Hashtable();
   for(int i=4; i<30; i=i+4){
// s3 = we.populatePredictUJ(sheetData1, listSimblances, i);
        s3 = we.populatePredictContentBoostedUJ(test1,
test2, Examples, listSimblances, i);
   double mae = we.getMAE1(test, original, s3);
                BigDecimal z1 = new
        BigDecimal(mae).setScale(2,BigDecimal.ROUND_H
        ALF_UP);
            mae = z1.doubleValue();
            System.out.println(" For neighbourset size -- " + i
+" MAE is " + mae);
            table.put(new Double(i), mae);
            }
```

6. Results

The MAE values are computed using Content Based CF and Content Boosted CF algorithms for different test data sets u1.test, u2.test, u3.test, u4.test and u5.test and tabulated in table 1 to table 5. MAE value for Content Based Predictor is also estimated and included in these tables for verification. The Comparative analysis of these computed values are presented in section 7.

6.1 MAE values for CF, CB and CBCF on U1.test dataset

Neighbor Set Size	4	8	12	16	20	24	28
MAE for CF	2.61	2.62	2.62	2.62	2.62	2.62	2.62
MAE for CB CF	0.98	0.91	0.89	0.86	0.86	0.85	0.85
MAE for CB CF	0.982.						

*Table.1. MAE values for different neighbor sets on **u1.test***

6.2 MAE values for CF, CB and CBCF on U2.test dataset

Neighbor Set Size	4	8	12	16	20	24	28
MAE for CF	2.61	2.61	2.61	2.61	2.61	2.61	2.61
MAE for CB CF	1.07	0.98	0.94	0.92	0.91	0.90	0.90
MAE for CB CF	1.02.						

*Table.2.MAE values for different neighbor sets on **u2.test***

6.3 MAE values for CF, CB and CBCF on U3.test dataset

Neighbor Set Size	4	8	12	16	20	24	28
MAE for CF	2.62	2.61	2.61	2.60	2.61	2.61	2.61
MAE for CB CF	1.07	0.98	0.95	0.94	0.93	0.92	0.91
MAE for CB CF	1.02.						

*Table3. MAE values for different neighbor sets on **u3.test***

6.4 MAE values for CF, CB and CBCF on U4.test dataset

.

Neighbor Set Size	4	8	12	16	20	24	28
MAE for CF	2.21	2.54	2.56	2.56	2.54	2.54	2.54
MAE for CB CF	1.11	1.05	1.01	0.99	0.99	0.98	0.97
MAE for CB CF	1.02.						

*Table4. MAE values for different neighbor sets on **u4.test***

6.5 MAE values for CF, CB and CBCF on U5.test dataset

Neighbor Set Size	4	8	12	16	20	24	28
MAE for CF	2.29	2.48	2.56	2.56	2.57	2.57	2.57
MAE for CB CF	1.13	1.04	1.01	1.00	0.99	0.98	0.97
MAE for CB CF	1.02						

*Table.5. MAE values for different neighbor sets on **u5.test***

7. Comparative Analysis

The MAE values tabulated in the above section are represented in bar diagrams for Comparative study. The different bar diagrams for different datasets U1.test to u5.test are presented in 7.1 to 7.5 sections respectively. From these diagrams, it can be easily noticed that CBCF shows the minimum MAE values for all datasets.

7.1. Bar diagram of MAE values for CF, CB and CBCF on U1.test dataset

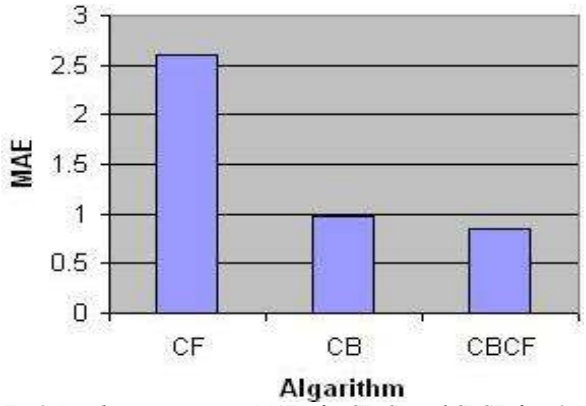


Fig.1: Bar diagram represent MAE's for CF, CB and CBCF for u1.test

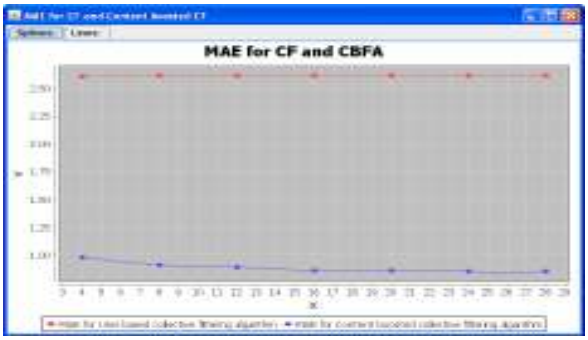


Fig 2: graphical representation of MAE values for CF and CBCF for different neighbor sets when run for u1.test.

7.2. Bar diagram of MAE values for CF, CB and CBCF on U2.test dataset

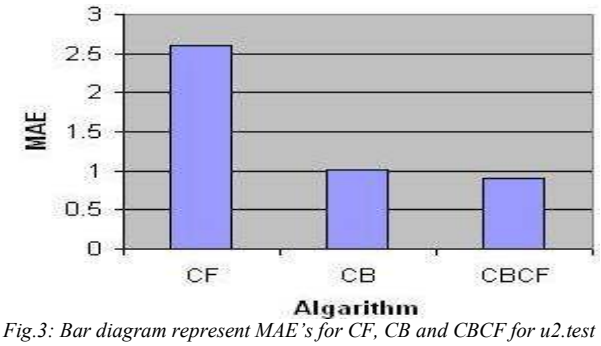


Fig.3: Bar diagram represent MAE's for CF, CB and CBCF for u2.test

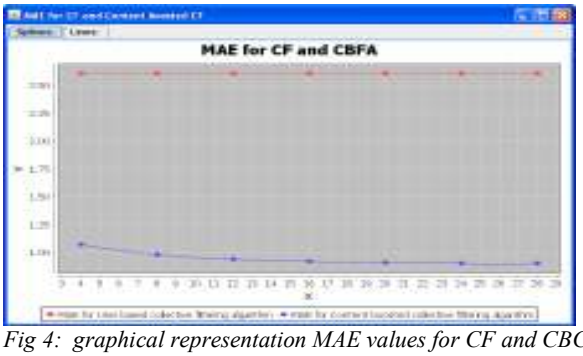


Fig 4: graphical representation MAE values for CF and CBCF for different neighbor sets when run for u2.test.

7.3. Bar diagram of MAE values for CF, CB and CBCF on U3.test dataset

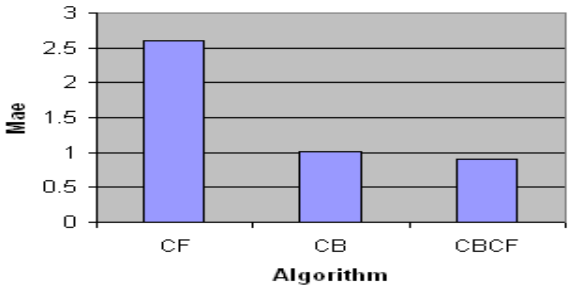


Fig.5: Bar diagram represent MAE's for CF, CB and CBCF for u3.test

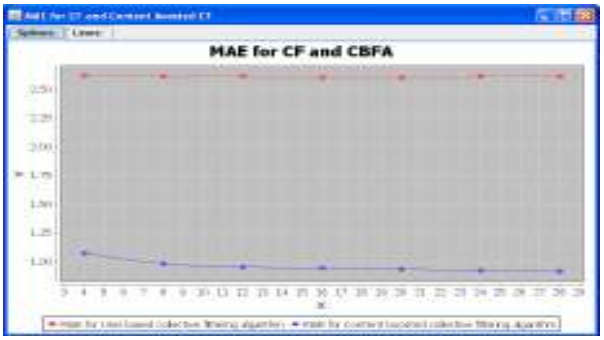


Fig 6: graphical representation MAE values for CF and CBCF for different neighbor sets when run for u3.test.

7.4. Bar diagram of MAE values for CF, CB and CBCF on U4.test dataset

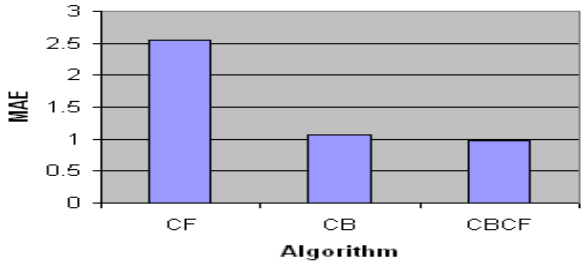


Fig.7: Bar diagram represent MAE's for CF, CB and CBCF u4.test

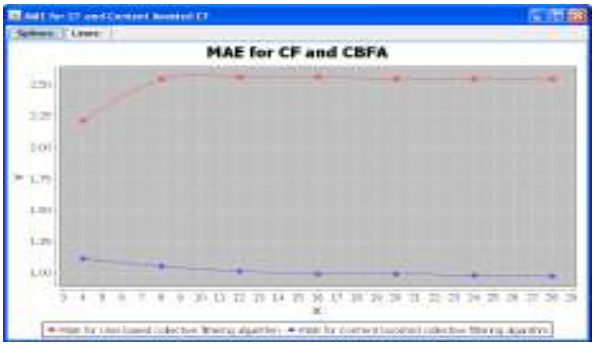


Fig 8: graphical representation MAE values for CF and CBCF for different neighbor sets when run for u4.test.

7.5. Bar diagram of MAE values for CF, CB and CBCF on U5.test dataset

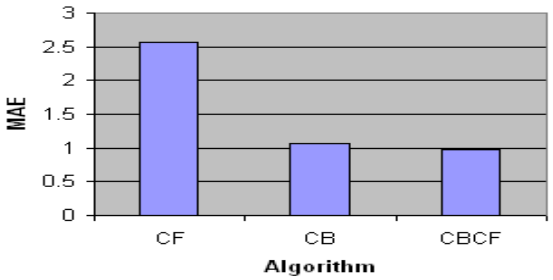


Fig.9: Bar diagram represent MAE's for CF, CB and CBCF for u5.test

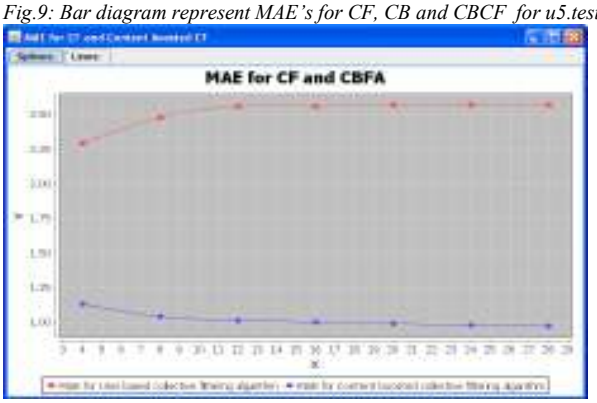


Fig 10: graphical representation MAE values for CF and CBCF for different neighbor sets when run for u5.test

7.6. Comparison of MAE values for CF, CB and CBCF:

A graph is drawn (Fig.11) by taking test datasets on x-axis and minimum MAE values for different algorithms on y-axis. The red line indicates the graph for CF predictor, the green line indicates the graph for content based CF and the blue line indicates the graph for content boosted CF. The graphical representation shows that the overall performance of the content boosted algorithm is better than the other Comparative algorithms

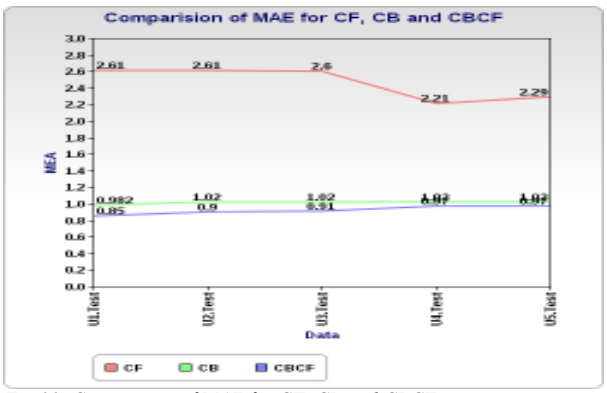


Fig.11: Comparison of MAE for CF, CB and CBCF

8. Conclusion

The derived results, comparative analysis and comprehensive study shows that Content boosted collaborative filtering algorithm puts forward for better performance among the other comparative algorithms. It is clearly understood that lesser MEA values indicates good recommendation quality. The results show that that CBCF is best and has good recommendation effect than the other algorithms.

9. References

[1] Xiuyan Gu, Linfeng Jiang, and Ziyi Zhang, "Study on User's Browse Behavior to Measure the User's Browse Interest", Network and Communication, vol.15, pp.43-45, 2005.

[2] Mustansar Ali Ghazanfar and Adam Pr¨ugel-Benne "Building Switching Hybrid Recommender System Using Machine Learning Classifiers and Collaborative Filtering" published in IAENG International Journal of Computer Science, 37:3, IJCS_37_3_09

[3] G. Adomavicius and A. Tuzhilin, "Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions", IEEE Transactions on Knowledge and Data Engineering 17 (2005), 634–749.

[4] R. Bell and Y. Koren, "Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights", IEEE International Conference on Data Mining (ICDM'07), pp. 43–52, 2007.

[5] Xiaoyuan Su and Taghi M. Khoshgoftaar "Review Article A Survey of Collaborative Filtering Techniques" Hindawi Publishing Corporation Advances in Artificial Intelligence Volume 2009, Article ID 421425, 19 pages doi:10.1155/2009/421425

[6] J. Bennet and S. Lanning, "The Netflix Prize", KDD Cup and Workshop, 2007. www.netflixprize.com.

[7] Z. Huang, D. Zeng and H. Chen, "A Comparison of Collaborative-Filtering Recommendation Algorithms for E-commerce", IEEE Intelligent Systems 22 (2007), 68–78.

[8] Lijuan Zheng,Yaling Wang, Jiangang Qi, Dan Liu "research and improvement of personalized recommendation algorithm based on collaborative filtering" published in IJCSNS International Journal of Computer Science and Network Security, VOL.7 No.7, July 2007

[9] Prem Melville and Raymond J. Mooney and Ramadass Nagarajan Content-Boosted Collaborative Filtering for Improved Recommendations published in Proceedings of the Eighteenth National Conference on Artificial Intelligence(AAAI-2002), pp. 187-192, Edmonton, Canada, July 2002

[10] R. Salakhutdinov, A. Mnih and G. Hinton, "Restricted Boltzmann Machines for Collaborative Filtering", Proc. 24th Annual International Conference on Machine Learning, pp. 791–798, 2007.

[11] R. Salakhutdinov and A. Mnih, "Probabilistic Matrix Factorization", Advances in Neural Information Pro-cessing Systems 20 (NIPS'07), pp. 1257–1264, 2008.

[12]movie lens data, http://movielens.umn.edu.

[13] group lens, http://www.grouplens.org/

[14] Bhaskar Mehta, Thomas Hofmann, and Wolfgang Nejdl, Robust Collaborative Filtering, In Proceedings of the 1st ACM Conference on Recommender Systems (Joseph A. Konstan, John Riedl, and Barry Smyth, eds.), ACM Press, October 2007, pp. 49–56.

[15] Bhaskar Mehta and Wolfgang Nejdl, Attack-resistant Collaborative Filtering, To appear: In Proceedings of the 31st ACM SIGIR Conference, ACM Press, 2008.

[16]amazon, an online portal, http://www.amazon.com/

[17] ebay, an online portal, http://www.ebay.in/

[18] netflix, an online video portal, https://www.netflix.com/

SPXS Sports Picks eXpert System

Luis A. Mateos
Automation and Control Institute (ACIN), Vienna University of Technology (TUW),
Gusshausstrasse 27 - 29 / E376, A - 1040 Vienna, Austria.
{mateos}@ acin.tuwien.ac.at

Abstract - *This paper presents a web based Sports Picks eXpert System (SPXS). The SPXS is an Expert System (ES) which knowledge is the database since 1990 of all the National Basketball Association (NBA) games and analysis of betting variables such as, against the spread points (ATS), total points line (TPL), difference of points; among others, from a team perspective. The SPXS try to mimic the sports handicappers, who give advice for selecting a specific team due to certain characteristics or properties of a game match. These virtual handicappers are represented as a knowledge-driven decision support system (DSS). Moreover, SPXS include an explanation of the logic behind the advice and transparency of the picking results. The system embeds several Artificial Intelligence (AI) prediction algorithms which represent different human handicapper experts, who have for priority different variables from the system, creating a group of experts competing for the best prediction results*.

Keywords: Expert system, Decision support system, Artificial intelligence, Prediction algorithm.

1 Introduction

One of the objectives of SPXS is to become a online development and testing platform for AI prediction algorithms in sports matches. In the same way, the system intent to develop fair betting lines, closer to binomial distribution values. Furthermore, the system purpose is to answer the question: Is it possible to predict in a deterministic manner sports matches results with a 54% or more certainty, in order for a bettor to have a profit?

SPXS latest version is a web based system, which includes a social network where users can login and gain access to the data base and features of the system, such as improved search engine and prediction algorithms. The system enables a user to track their predicted results against the systems experts (AI prediction algorithms) and other users.

One of the system main features is the way the main results are presented. The results are modeled as a combination of binary-ternary variables, arranged as DNA tables. Which from a human perspective, the results are easier to analyze. It is easier to analyze colors in order to differentiate results, than pages filled with numeric data.

A Sports Picks eXpert System (SPXS) is presented in this paper. Section II gives a brief description of the problem domain, rules and common practices. Section III presents the proposed expert system SPXS, describing its main features and design. Section IV present a case of study, where by modeling the results as a finite state machine (FSM), the system is able to find tendencies of teams over seasons.

1.1 Expert Systems

One of the goals of Artificial Intelligence is to develop systems which exhibit 'intelligent' human-like behavior. Initial attempts in the field (in the 1960s) like the General Problem Solver created by Allen Newell and Herbert Simon from Carnegie Mellon University, were to create a general purpose intelligent systems which could handle tasks in a variety of domains. However, researchers soon realized that developing such general purpose systems was too difficult and it was better to focus on systems for limited domains. Edward Feigenbaum from Stanford University then proposed the notion of expert systems. Expert systems are systems which encode human expertise in limited domains and can represent this human knowledge using If-then rules. [3].

Expert systems can be created almost for any domain for which there exists a human expert. Some of the expert systems which have been created are:
DENDRAL – Considered being the first expert system. Identifies the molecular structure of unknown compounds; developed by Stanford University.
PREDICTE – Given information about a high-rise building to be constructed, it provides estimates of the time required to construct it; developed by Digital Equipment Corporation for use by Land Lease, Australia.

An expert system can be defined as "an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solutions [4]". This means that human expertise can be transferred to a computer program, which is able to infer and make conclusions, giving the expert advice of the topic and if necessary explains the logic behind the advice.

Expert systems provide powerful and flexible means for obtaining solutions to a variety of problems that often cannot be dealt with by other, more traditional and orthodox methods [5]. The term knowledge-based system (KBS) is often referring as expert systems.

The four main components of KBS are: a knowledge base, an inference engine, a knowledge engineering tool, and a specific user interface. Some of KBS important applications

include the following: failure analysis, knowledge representation, decision support/making and learning.

The concept of a decision support system (DSS) can be broadly defined as a class of computerized information systems that support decision-making activities. Turban, defines it as "an interactive, flexible, and adaptable computer-based information system, especially developed for supporting the solution of a non-structured management problem for improved decision making. It utilizes data, provides an easy-to-use interface, and allows for the decision maker's own insights" [7].

A Knowledge-driven DSS can suggest or recommend actions based on specialized problem-solving expertise stored as facts, rules, procedures, or in similar structures. Additional knowledge is handled through data mining tools and intelligent methods [10].

2 Expert sports handicapping

In sports betting, the major sports such as NBA and NFL use point spread as a form of wagering. The point spread is an adjustment to the final score made by the bookmakers in order to keep the same chance for both teams to *win/lose* the bet from a game match.

2.1 Odds, spread points and lines

The best way to explain the variables in consideration for sport betting is with an example; the point spread on a NBA game between the Los Angeles Lakers and the Miami Heat is as follows:

Los Angeles Lakers	190	
Miami Heat	-2	(-110)

The point spread in this NBA game is -2 to the Heat. This means that if you bet the spread on the Heat, Miami need to win the game by 3 points or more in order to win the bet. It is common to call this feature ATS, means "against the spread", so the Heat are 2 points ATS. If Miami wins by 2 points, the bet is "no action" and none wins or losses.

From the above example, if the final score is Los Angeles 90 and Miami 96, the Heat cover the point spread, since they won the game by 4 points ($4 > (ATS=2)$), and if the final score is Miami 93 – Los Angeles 92, the Heat fail to cover the spread ($1 < (ATS=2)$) and the bet is lose. If the Heat loses the game, automatically the bet is lost, no matter the final score. The other case is for Los Angeles, means they can lose by no more than 2 points in order to cover (win) the bet, even if Los Angeles loses, as long as they lose by less than 2 points.

On the above example, the numbers in the brackets are the payout odds of the match. Generally the pays out is -110 (i.e. you have to bet \$110 to win \$100 + the \$100 bet). From this relation the percent of winning/losing bets need to be at least 54.5% in order to obtain a 50 – 50 relation.

The team expected to win the game (the "better" team) is called the "favorite" and is the one with the "minus" sign in front of the point spread. The "weaker" team gets the + point spread, gets extra points added to the final score.

From the same example the number 190 indicate the Total Points Line (TPL); a bet can be made to *over* or *under* this quantity. *Team A score + Team B score = Total points*, if $A+B > TPL$ then the winning bet is over, if $A+B<TPL$ the winning bet is under, if $A+B=TPL$ it is a *push*.

2.2 Handicapping and fair point spread

The points spread objective, is that both bets; for the favorite team and for the not favorite team, are equally probable to cover the bet, 50-50 percent. In the same way, the TPL value is set, in order to mimic a perfect binomial distribution.

SPXS is able track the differences from the ATS and TPL and detected irregularities from the proposed points spread lines given by the sports books. In the same way human handicappers do, SPXS search for patterns and irregularities. Constant patterns over time are used to predict the next result, following these constant behaviors. While, for irregularities, is expected variables to change, given their instability.

3 SPXS

Sports Picks eXpert System objectives are: to become a platform for development/testing AI prediction algorithms and to be a Knowledge Based-System able to recognize patterns and infer tendencies of the teams.

3.1 SPXS – System design

The technology used by the latest SPXS is a combination of Linux, Apache, MySQL and PHP (LAMP), in order for the systems to be online and free access via internet. Previously, the alpha version of SPXS was done in Delphi 4.0, this version was a standalone application [1].

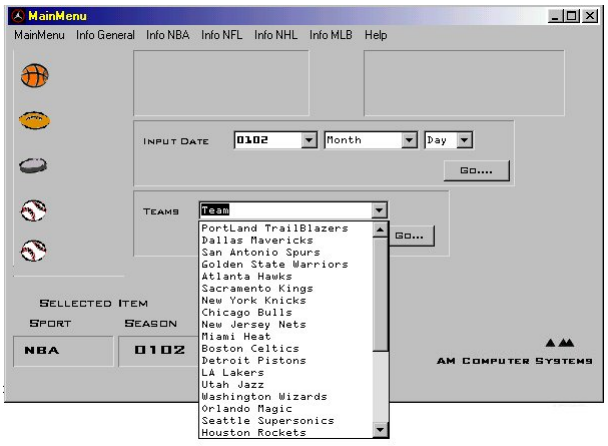


Fig. 1. SPXS – alpha version, standalone application, including the four major American sports.

3.2 SPXS – data base (dB) fields

The design of the dB consists on a primary unique key represented by the teams plus the match date, which define a unique game match. The fields/variables are divided in groups: A *team(s)* variable group, to select a specific team or all the teams. A group of variables forming the season/date fields, such as *month, day, year, season, type of game* (regular or playoff). Another group, representing the scores from both teams and properties of the match, i.e., played in *over time* (OT). The left groups focus on the analysis of the scores and betting variables, such as the *odds, ATS, total points line, the winner, the team that covers the bet, the difference of winning points, bet points and total points*.

Additionally, to enable a faster query, there are tables per each *team* and a table including all *teams*. So the queries involving all teams are accessible with only one table, instead of searching information from different tables.

In the system, the main search form consists of 9 searching variable groups. Each group can be expanded in order to specify the search parameters, as shown in figure 2.

Fig. 2. SPXS - dB search Graphical User Interface (GUI).

A search may consist on a maximum of 30 variables, enabling the query to find highly specific information and in the same way enabling the system to find relevant patterns.

3.3 SPXS - DNA visualization

SPXS includes a visualization feature to easily review the main results from a query for a human user. The main three results are represented as a DNA tables with different colors, black, white and blue. The main three results from a game match are: WL *win/lose*; ATS (against the spread) *cover/no cover/push*; TPL (total points line) *over/under/push*.

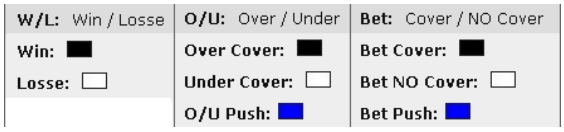


Fig. 3. SPXS – DNA codes information.

An example of the DNA visualization for a team search is shown in figure 9, where the results are presented as a table with the main variables as DNA codes and the extended information is given in numeric form.

The system includes predefined tables named DNA Xtatistics, where the DNA visualization and resumed information is presented for a team or group of teams. This is a concise representation of all the system variables in order to notice any tendency from a team. It also takes into account where the team played, at *home* or *away*, as shown in fig. 4.

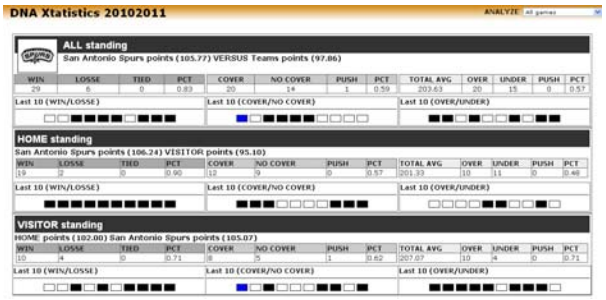


Fig. 4. SPXS - DNA Xtatistics, concise visual information.

3.4 SPXS - AI prediction algorithms

The AI prediction algorithms included in SPXS varies from the simple points average to complex ones, which takes into account previous results, as well as straight winning or losing games, relating the obtained information to previous seasons and tendencies of the teams.

The basic AI prediction algorithm consists in the relation of three simpler algorithms [12]:

Average algorithm

This basic algorithm takes the average of points per each team, taking into account if a team is playing as *local* or *away*, resulting in the predicted scores.

Difference points algorithm

This algorithm takes into account the difference of points, from the difference by the *home-away* averages for each team, resulting in the ATS.

Home-Away by team algorithm

This algorithm set a total points line (TPL) from the average of the team's total points.

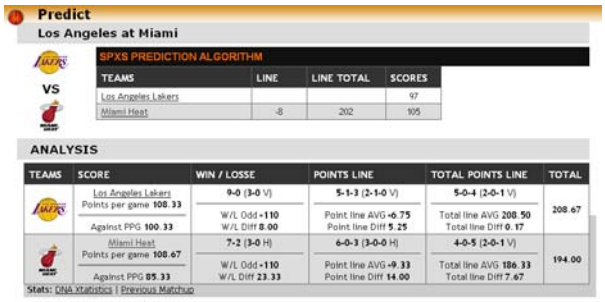


Fig. 5. SPXS - Basic AI prediction algorithm.

3.5 SPXS – Finite State Machine

A finite-state machine (FSM), is a behavior model composed of a finite number of states, transitions between

those states, and actions, similar to a flow graph in which it is possible to inspect the way logic runs when certain conditions are met [6]. It has finite internal memory, an input feature that reads symbols in a sequence, one at a time without going backward; and an output feature, which may be in the form of a user interface, once the model is implemented [8]. The operation of an FSM begins from one of the states (called a start state), goes through transitions depending on input to different states and can end in any of those available, however only a certain set of states mark a successful flow of operation (called accept states) [9].

A SPXS-FSM algorithm was developed in order to find the tendency over the league seasons (1990-1991 to 2009-2010) for each team, similar to a finite state machine. Where the state values of the machines are the extended three main variables W/L (*win/lose*), ATS (*cover/no cover/push*) and TPL (*over/under/push*), dismissing the neutral *push* states. Therefore, a FSM is computed for each of the six variables: *win* | *lose* | *cover* | *no cover* | *over* | *under*.

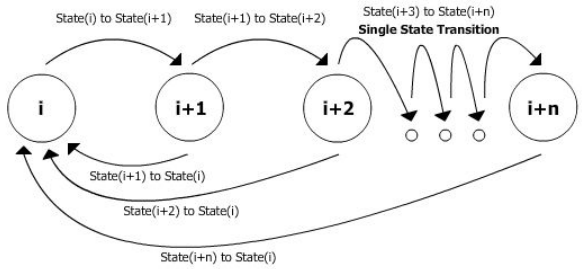


Fig. 6. SPXS - Finite-state machine model.

The FSM model, work as follows: If the team in consideration initial state is a *win*, the machine WIN saves the *n wins* until a *lose* state appears, in this case the machine LOSE starts to count and saves the *n* states until a *win* appear and so on. In the same way, the machine BET COVER and machine BET NO COVER saves the *n covers* and *n no covers* state depending on the initial and the following states, respectively. The machine OVER and machine UNDER saves the *n overs* and *n unders* states that follows the starting state and following states respectively, as shown in figure 6.

4 Case of study – Team tendency

Among the Sports Picks eXpert System objectives, one is to become a Knowledge Based-System (KBS) able to recognize patterns and infer tendencies of the teams from its current and previous states. By processing the SPXS-FSM algorithm, it is possible to obtain patterns and tendencies from teams over the season's league results, for the six main variables.

Theoretically, the resulting values from the FSM should be the same for each couple of machines: ATS (BET COVER/BET NO COVER) and TPL (OVER/UNDER), except machines W/L, which results are not compensated. Ideally, the results from each FSM should be represented by a perfect binomial distribution. Nevertheless, in practice the results vary considerable.

The binomial distribution is the discrete probability distribution of the number of successes in a sequence of *n* independent *yes/no* experiments, each of which yields success with probability *p*.

In figure 7, the results from SPXS-FSM for the season 2009-2010 for the Utah Jazz are shown. The variable to watch in this example is the machine BET NO COVER, which shows big variations from its binomial distribution. For a BET NO COVER to continue in the same state after an initial bet no cover n=1, it happens 6 times out of 22, resulting p=27.27%; from this second state n=2 for the same state to continue to n=3, the value change to 2 times out of 6, resulting p=33.33%. The binomial distribution is completely lost, showing an irregularity from the sports books line makers, who set the ATS. The other three FSM show a closer distribution of results to a binomial values, the machine COVER show 10,4,2,4,0,0,1; OVER 11,7,4,0,0,1; UNDER 14,6,2,0,1.

Utah Jazz — Season 20092010

Score Visitor	Score Local	Odd & difference W/L	POINTS Line & difference	TOTALS Line & difference	Total
TOTAL 8157	8496	-9020 | 435	-303 | 132	16698 | -45	16653
AVERAGE 99.48	103.61	-110.00 | 5.30	-3.84 | 1.67	208.73 | -0.56	203.09

SUMMATORY	DNA CODE	Win/Lose	Cover/NoCover	Over/Under
OVER ALL				
WIN / COVER / OVER		53	49	43
LOSE / NO COVER / UNDER		29	30	37
TIED / PUSH / PUSH		0	3	2

WIN

No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Win(n)	11	4	3	4	0	0	0	0	1	0	0	0	0	0	0
Win(inf - n)	12	8	5	1	1	1	1	1	0	0	0	0	0	0	0
Win(inf - n)/(n)	1.09	2.00	1.67	0.25	1	1	1	1	0	0	0	0	0	0	0
Win(inf - n)/(n)	■	■	■	□	■	■	■	■							

LOSE

No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Lose(n)	20	3	1	0	0	0	0	0	0	0	0	0	0	0	0
Lose(inf - n)	4	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Lose(inf - n)/(n)	0.20	0.33	0	0	0	0	0	0	0	0	0	0	0	0	0
Lose(inf - n)/(n)	□	□													

BET COVER

No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Cover(n)	10	4	2	4	0	0	0	0	1	0	0	0	0	0	0
Cover(inf - n)	11	7	5	1	1	1	1	1	0	0	0	0	0	0	0
Cover(inf - n)/(n)	1.10	1.75	2.50	0.25	1	1	1	1	0	0	0	0	0	0	0
Cover(inf - n)/(n)	■	■	■	□	■	■	■	■							

BET NO COVER

No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
NOCover(n)	16	4	2	0	0	0	0	0	0	0	0	0	0	0	0
NOCover(inf - n)	6	2	0	0	0	0	0	0	0	0	0	0	0	0	0
NOCover(inf - n)/(n)	0.3	0.50	0	0	0	0	0	0	0	0	0	0	0	0	0
NOCover(inf - n)/(n)	□	□													

OVER

No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Over(n)	11	7	4	0	0	1	0	0	0	0	0	0	0	0	0
Over(inf - n)	12	5	1	1	1	0	0	0	0	0	0	0	0	0	0
Over(inf - n)/(n)	1.09	0.71	0.25	1	1	0	0	0	0	0	0	0	0	0	0
Over(inf - n)/(n)	■	□	□	■	■										

UNDER

No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Under(n)	14	6	2	0	1	0	0	0	0	0	0	0	0	0	0
Under(inf - n)	9	3	1	1	0	0	0	0	0	0	0	0	0	0	0
Under(inf - n)/(n)	0.64	0.50	0.50	1	0	0	0	0	0	0	0	0	0	0	0
Under(inf - n)/(n)	□	□	□	■											

Fig. 7. SPXS – Utah Jazz 2009-2010 season FSM analysis.

Ideally the results from a binomial distribution are:

$$\text{Results} = R/2^1 + R/2^2 + R/2^3 + \ldots + R/2^n \quad (1)$$

For example, if the total results are R=63, the perfect binomial distribution is: 32 for *n*=1, 16 for *n*=2, 8 for *n*=3, 4 for *n*=4, 2 for *n*=5, 1 for *n*=6.

The presented SPXS-finite state machine results are from a single season, 2009-2010. SPXS system contains the results since the year 1990. Thus, the SPXS-FSM was modified to obtain the machines values for the entire data base, corresponding to 20 seasons, from 1990-1991 to 2009-2010.

In figure 8, the final values from the finite state machine BET NO COVER are shown. For a season to obtain an initial value of BET NO COVER, it is *p*=52.63%; this is a good approximation to the binomial distribution. Nevertheless, for a season, from a starting value of BET NO COVER at *n*=1 to change in *n*=2, it is *p*≈75%.

This shows that over the seasons, there are tendencies from team to team, which can be estimated. Thus, the betting lines can be improved to generate a fair betting system closely to the intended 50 – 50 percent binomial distribution. The results show that SPXS is a tool to analyze and predict results for sports game matches. Moreover SPXS is able to find pattern from the dB by modeling the main variables as a FSM.

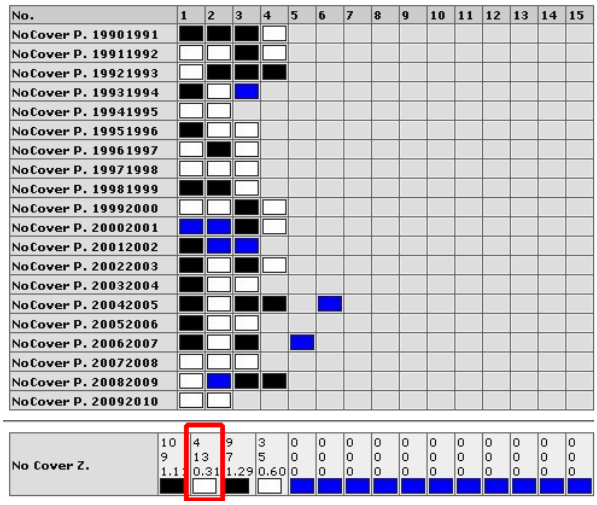


Fig. 8. SPXS – Utah Jazz BET NO COVER Finite State Machine, DNA values for the seasons 1990-1991 to 2009-2010.

In the same way, these patterns over seasons identify periods when the team in consideration was a dominant or a building up franchise. Browsing the SPXS finite state machines for Chicago bulls, it is possible to notice the Jordan and the after Jordan seasons, as well as other periods involving dominant player in different teams.

5 Conclusions

This paper presents an expert system able to predict game results and analyze the betting variables from the National Basketball Association (NBA) games. The Sports Picks eXpert System (SPXS) knowledge data base consists of all NBA games since 1990.

Basically, the system is a social network and a platform for development and test of AI prediction algorithms in NBA game matches. The system includes several AI prediction algorithms, mimicking sports handicappers, who give advice for selecting a specific team due to certain characteristics or properties of a game match. These virtual handicappers are represented as a knowledge-driven decision support system (DSS), including an explanation of the logic behind the advice and transparency of the picking results.

The SPXS user interface includes a novel way to review sports betting data. Instead of presenting the results with only numeric data, DNA tables are used to represent the three main betting variables of the system (*win/lose*; from against the spread (ATS) *cover/no cover/push*; from total points line (TPL) *over/under/push*) for the user to easily notice patterns or inconsistencies in the data.

The expert system can be modeled as a finite state machine per each of its main variables, revealing notable differences from its expected binomial distribution. Hence, the system is able to find patterns over a team in a season or over seasons, giving an extra edge to the AI prediction algorithms.

5.1 Future work

Currently, the system data base consist only, in the NBA games results and analysis since 1990 to the current 2010-2011 season [2]. The long term goal for SPXS is to include all four major American sports, NBA (National Basketball Association), NFL (National Football League), MLB (Major League Baseball) and NHL (National Hockey League) as originally designed [1]. In the same way, the possibility for the system to expand to other leagues. Since, SPXS latest version is a web based system, capable to become an open source platform, where users will be able to improve and upgrade the system.

6 References

[1] Luis A. Mateos. "Sistema Experto Sports Picks". Master Thesis in Computer Science ITESM campus GDL (2002). http://www.amcomputersystems.com/AM/research/SPXS_MCC_LuisAMateos.pdf.

[2] Luis A. Mateos. "SPXS Sports Picks eXpert System online platform". AM Computer Systems (2011). http://www.amcomputersystems.com/spxs/.

[3] K S R Anjaneyulu. "Expert Systems: An Introduction"; Research Scientist in the Knowledge Based Computer Systems Group at NCST. Resonance article (1998).

[4] Joseph Giarratano, Gary Riley. "Expert Systems: Principles and Programming", Fourth Edition (2004).

[5] Shu-Hsien Liao. "Expert system methodologies and applications - a decade review from 1995 to 2004", Expert Systems with Applications, 28, 93-103 (2005).

[6] Wagner, F. "Modeling Software with Finite State Machines: A Practical Approach". Auerbach Publications, 2006, ISBN 0-8493-8086-3.

[7] Turban, E. "Decision support and expert systems: management support systems". Englewood Cliffs, N.J., Prentice Hall.

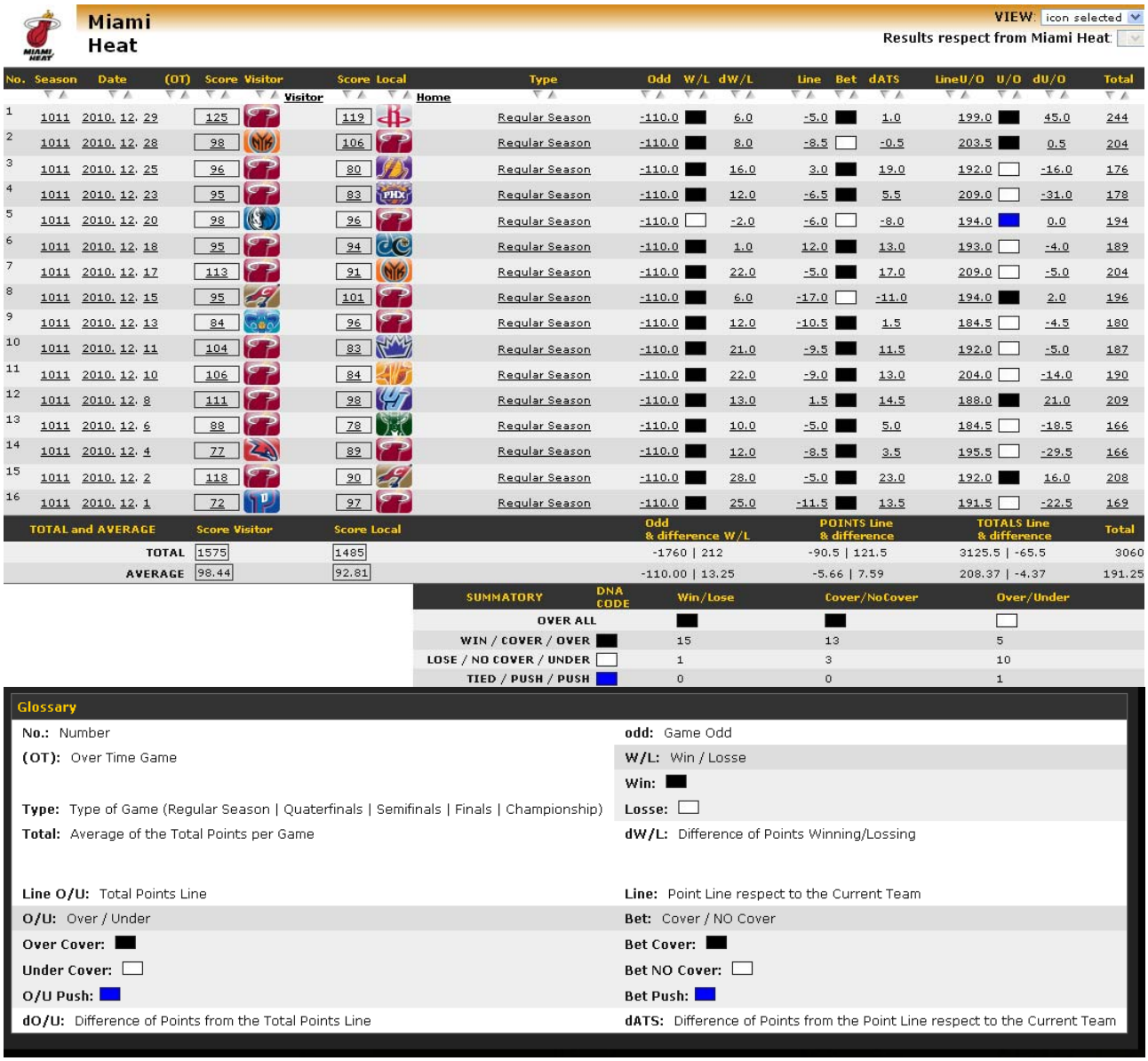
[8] Samek, M. "Practical State charts in C/C++". CMP Books, 2002, ISBN 1-57820-110-1.

[9] Sipser, Michael. "Introduction to the Theory of Computation" (2nd ed.). Boston Mass: Thomson Course Technology (2006). ISBN 0-534-95097-3.

[10] Gebus, S. "Knowledge-based Decision Support systems for Production Optimization and Quality Improvement in the Electronics Industry". Oulu University Press (2006). ISBN 951-42-8205-1.

[11] Bernhard Nebel. "On the compilability and expressive power of propositional planning formalisms". Journal of Artificial Intelligence Research, 12:271–315, (2000).

[12] Luis A. Mateos. "SPXS Sports Picks eXpert System Odds algorithms". AM Computer Systems 2011. http://www.amcomputersystems.com/spxs/odds_algorithm.



Miami Heat

VIEW: icon selected
Results respect from Miami Heat:

No.	Season	Date	(OT)	Score Visitor	Visitor	Score Local	Home	Type	Odd	W/L	dW/L	Line	Bet	dATS	LineU/O	U/O	dU/O	Total
1	1011	2010. 12. 29		125		119		Regular Season	-110.0		6.0	-5.0		1.0	199.0		45.0	244
2	1011	2010. 12. 28		98		106		Regular Season	-110.0		8.0	-8.5		-0.5	203.5		0.5	204
3	1011	2010. 12. 25		96		80		Regular Season	-110.0		16.0	3.0		19.0	192.0		-16.0	176
4	1011	2010. 12. 23		95		83		Regular Season	-110.0		12.0	-6.5		5.5	209.0		-31.0	178
5	1011	2010. 12. 20		98		96		Regular Season	-110.0		-2.0	-6.0		-8.0	194.0		0.0	194
6	1011	2010. 12. 18		95		94		Regular Season	-110.0		1.0	12.0		13.0	193.0		-4.0	189
7	1011	2010. 12. 17		113		91		Regular Season	-110.0		22.0	-5.0		17.0	209.0		-5.0	204
8	1011	2010. 12. 15		95		101		Regular Season	-110.0		6.0	-17.0		-11.0	194.0		2.0	196
9	1011	2010. 12. 13		84		96		Regular Season	-110.0		12.0	-10.5		1.5	184.5		-4.5	180
10	1011	2010. 12. 11		104		83		Regular Season	-110.0		21.0	-9.5		11.5	192.0		-5.0	187
11	1011	2010. 12. 10		106		84		Regular Season	-110.0		22.0	-9.0		13.0	204.0		-14.0	190
12	1011	2010. 12. 8		111		98		Regular Season	-110.0		13.0	1.5		14.5	188.0		21.0	209
13	1011	2010. 12. 6		88		78		Regular Season	-110.0		10.0	-5.0		5.0	184.5		-18.5	166
14	1011	2010. 12. 4		77		89		Regular Season	-110.0		12.0	-8.5		3.5	195.5		-29.5	166
15	1011	2010. 12. 2		118		90		Regular Season	-110.0		28.0	-5.0		23.0	192.0		16.0	208
16	1011	2010. 12. 1		72		97		Regular Season	-110.0		25.0	-11.5		13.5	191.5		-22.5	169

TOTAL and AVERAGE	Score Visitor	Score Local	Odd & difference W/L	POINTS Line & difference	TOTALS Line & difference	Total
TOTAL	1575	1485	-1760 | 212	-90.5 | 121.5	3125.5 | -65.5	3060
AVERAGE	98.44	92.81	-110.00 | 13.25	-5.66 | 7.59	208.37 | -4.37	191.25

SUMMATORY	DNA CODE	Win/Lose	Cover/NoCover	Over/Under
OVER ALL				
WIN / COVER / OVER		15	13	5
LOSE / NO COVER / UNDER		1	3	10
TIED / PUSH / PUSH		0	0	1

Glossary

No.: Number
(OT): Over Time Game
Type: Type of Game (Regular Season | Quaterfinals | Semifinals | Finals | Championship)
Total: Average of the Total Points per Game
Line O/U: Total Points Line
O/U: Over / Under
Over Cover:
Under Cover:
O/U Push:
dO/U: Difference of Points from the Total Points Line
odd: Game Odd
W/L: Win / Losse
Win:
Losse:
dW/L: Difference of Points Winning/Lossing
Line: Point Line respect to the Current Team
Bet: Cover / NO Cover
Bet Cover:
Bet NO Cover:
Bet Push:
dATS: Difference of Points from the Point Line respect to the Current Team

Fig. 9. SPXS - DNA visualization among numeric information, DNA values described in the glossary of the page.

An Intelligent Method for Retrieval of Verbal Terms from the Web as Answers in Response to Complex Interrogative Sentences

***Hirokazu Watabe,**

Dept. of Intelligent Information Engineering and Science,
Doshisha University, Kyotanabe, Kyoto, 610-0394, Japan
hwatabe@mail.doshisha.ac.jp

Misako Imono, Eriko Yoshimura and Seiji Tsuchiya

Dept. of Intelligent Information Engineering and Science,
Doshisha University, Kyotanabe, Kyoto, 610-0394, Japan
mimono@indy.doshisha.ac.jp, eyoshimura@indy.doshisha.ac.jp, stsuchiy@mail.doshisha.ac.jp

Abstract - *In this study, we propose an intelligent method for automatic retrieval of terms from the Web as answers (in word form) to questions (in interrogative sentence form), that utilizes a concept base that expresses the semantic features of terms in both words and weightings. This provides a basis for the selection of answers from Web documents, together with calculation of numerical values representing the semantic association between terms and node scoring, and allows for the flexible selection of the answers that will be returned in response to the queries.*

Keywords: WWW, question answering, concept base, degree of association

1 Introduction

With existing search systems, it is increasingly difficult to locate and retrieve information limited to the scope required by the user. This is due primarily to the rapid expansion of the Web in both its reach and its user numbers, as well as the enormous cumulative volume of its online digitized documents. As a result, desired information is often buried in other concurrently retrieved information. Thus, new techniques are needed to facilitate the accurate, focused retrieval of the information sought by the user. Question answering (QI) is a particularly promising solution, but it generally requires stepwise analysis and the registration of knowledge into a database. Furthermore, processing the queries presented in interrogative form poses a particular difficulty due to the high degree of freedom of expressions that can be used in such queries and in Web documents.

In response to these issues, we propose an intelligent method for automatic retrieval of verbal answers from the Web, with emphasis on proper nouns, to such questions. To accomplish this, the system must be able to identify the type of answer that is being sought by the interrogative, and then assess the suitability of the retrieved answers. For answers consisting of proper nouns, it is particularly difficult to gauge whether they are appropriate to the type of answer being sought. This paper explores a method, referred to as intelligent Web retrieval (iWR), which addresses this issue. Our method is not wholly dependent on word notation. Instead, it incorporates a flexible association capability through its use of a "concept base"[1] and a "degree of association calculation"[2] association mechanism, together with "question semantics understanding", "undefined term attributes acquisition"[3], "optimal node determination"[4], and "scoring calculation" components – all of which are combined for enhanced utilization of Web information. Using these, we could develop a system that enables flexible answer selection.

2 Overview of intelligent Web retrieval system

Figure 1 shows the basic components of the iWR system designed for selection of answers to interrogative questions addressed to the Web by the user. For example, if the question is ("Where were the 2008 Olympics held?"), the objective is then to output the correct answer ("Beijing").

The three main component processes in the iWR system are question analysis, candidate answers acquisition, and answer scoring. Question analysis incorporates a question semantics understanding function that can acquire query target terms (i.e., terms being sought by the query) and an optimum node determination function. Answer scoring incorporates a term attributes acquisition function and a node scoring calculation function used for optimum node

determination. The overall system also incorporates a concept base consisting of a large-volume knowledge base constructed mechanically from multiple Japanese-language dictionaries, newspapers and other sources, as well as a degree of association calculation function (an association mechanism) that quantitatively assesses the strength of association between concepts. The system is capable of highly flexible term acquisition from the Web, which generally contains multiple word variants, so it can be used with existing search engines [7].

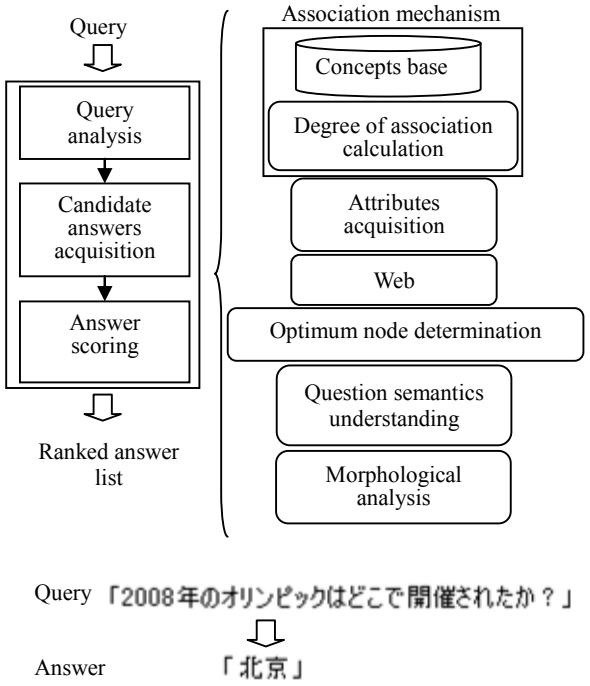


Figure 1: Basic concept of the intelligent Web retrieval system.

3 Association mechanism and question semantics understanding function

3. 1 Concept-Base

A concept-base is a knowledge base comprised of terms (concepts) mechanically constructed from sources such as multiple Japanese language dictionaries and newspapers and terms (attributes) that express their semantic features. Concepts have been given attributes along with a weighting, which expresses their importance. Approximately 120,000 concept notations have been compiled in the concept base, with an average of 30 attributes for one concept. A certain concept A has a pair of sets of attribute a_i and weighting w_i, as appear below.

$$A = \{(a_1, w_1), (a_2, w_2), \cdots, (a_m, w_m)\}$$

Any primary attribute a_i is composed of the terms contained in the set of concept notations in its concept base. Therefore, to ensure that a primary attribute matches a certain concept notation, that primary attribute can be further extracted. This is called a secondary attribute. In a concept base, a concept is defined by a chained set of attributes to the *n*-th order.

3.2 Degree of Association

Each concept is defined as a set of attributes and each attribute is also a concept as described above. In this paper, a method to derive the degree of association between concepts using up to second order attributes is used. The value of the degree of association ranges from 0.0 to 1.0.

3.3 Question semantics understanding function

The question semantics understanding function employs a parsing tool to acquire query target terms from the question. If, for example, the question contains a question-word expression such as “誰” (“who”) or “どこ” (“where”), this function will then acquire “person” or “place” as a query target term. It can also acquire query target terms from questions containing the question word “何” (“what”) as well as from questions in which the question word itself is omitted.

4 Web information utilization techniques[3,4]

Concepts not registered in the concepts base are referred to as “undefined terms”. Although semantic expansion, and thus degree of association calculation, cannot be performed for such terms, attributes and weights can be assigned to them by the application of an undefined term attributes acquisition technique [3]. Throughout this paper, we refer to the process of assigning a set of attributes and a weight to a given term as “conceptualization”. The optimum node determination technique [4] is employed to select the appropriate “answer type” from among various “candidate answer types”, as described in Sec. 5 below.

5 Candidate answer types

The assessment of what kind of answer is being sought by the question will strongly influence the answer selection. For example, if the question is “….はどこか” (“Where is….”), it will be interpreted to mean that a “場所” (”place”) is expected as the answer, and it is accordingly necessary to determine whether the candidate answers are appropriate for this category. In this paper, we refer to the answer category being sought by the question as the “answer type”, which is to be determined by selection from the “candidate answer types” (Table 1).

Table 1: Candidate answer types.

Candidate answer type	Typical target term
Person	Name of person
Organization	Company or university
Place	Place name, country name
Animal	Name of animal
Plant	Name of plant
Substance	Name of substance (natural)
Product	Name of product (man-made)
Work	Name of work (man-made)
Event	Name of event
Incident	Name of incident
Language	Name of language
System	Name of system

The twelve candidate answer types (Table 1) have been chosen based on prevalence and facility. Those such as person, place, and organization may be readily inferred from the question words that frequently occur in queries, such as "who" ("誰") and "where" ("どこ"). Those such as "product" and "work" (work of art, literature, music, etc.) are terms of semantic taxonomy that readily serve as Web search targets. In this paper, the type of question targeting the answer is taken as the candidate answer type.

6 Intelligent Web retrieval process

The process flow of the iWR system is essentially as shown in Fig. 2.

Upon question input, question analysis is performed to determine the parameters (answer types and search terms) of the search for the answer. The search terms are then input to obtain candidate answers from the Web. Next, search term and candidate answer conceptualizations are performed in preparation for the answer scoring process. The candidate answers are then scored and output in the order of their score rankings, from high to low.

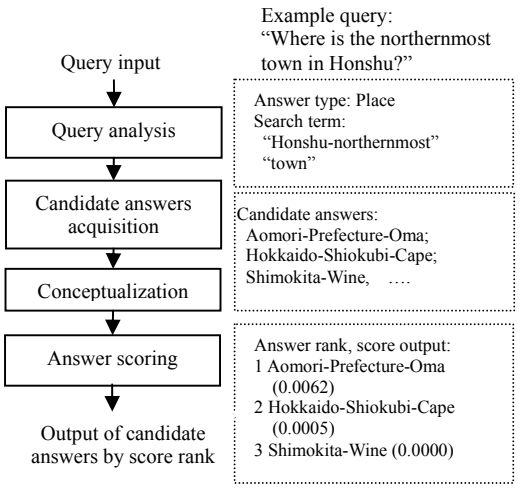


Figure 2: Intelligent Web retrieval (iWR) process flow.

6.1 Question analysis

6.1.1 Determination of answer type

The question is input to the question semantics understanding component to acquire the query target term, which provides the basis for determining the answer type. In essence, this determination is performed by using a previously constructed knowledge base to identify the answer type to which the query answer belongs. Table 2 shows three sample queries and their corresponding answer types. If the query target term is not present in the knowledge base, then it is subjected to further processing.

If the query target term is "person", then "person" is taken as the answer type. If the query target term is "place", then the degrees of association between that term and the candidate answer types "place" and "organization" are calculated, and the candidate answer type with the higher degree of association is taken as the answer type. If a query target term other than "person" or "place" is input, it is subjected to the answer-type determination processing and the resulting answer-type candidate is taken as the answer type.

Table 2: Typical answer type determination.

Question	Answer type
Who founded Doshisha Academy, the predecessor of Doshisha University?	Person
Where is the northernmost town in Honshu?	Place
What is the international movie festival that is held every five years in France?	Event

6.1.2 Acquisition of search terms

Morphological analysis of the question is performed using the ChaSen morphological analyzer[6], and the substantive terms and compound terms (alphanumeric series) are extracted as question keywords (excluding verbs and adjectives), which are linked by an intervening space and then taken as the search term (Table 3).

In standard Japanese orthography, no space is ordinarily used between words. In the English translations shown below, this special "intervening space" is indicated by an ellipsis (…) and the absence of a space between words is indicated by a hyphen (-).

Using these steps, the question content is extracted as keywords to form search terms that can be used effectively. In principle, this can be applied to any question (i.e., sentence in interrogative form), whether it is a simple or compound sentence.

Table 3: Typical search terms acquisition.

Question	Search term
"本州最北端の町はどこか?" (Where-is-the-northernmost-town-in-Honshu?)	"本州最北端　町" "Honshu-northernmost...town"

6.2 Acquisition of candidate answers

Candidate answers to the query are acquired from the Web by the following procedure:

1) The search term is input and a search is performed with the search engine. Then, the contents of the 100 top-ranking search result pages are acquired.
2) For this document corpus, HTML tags and other unnecessary information are removed and morphological analysis[6] is performed to acquire their substantive terms.
3) Compound terms are acquired for the morphemes, using the condition "noun, number, alphabet series are compounds".
4) For the extracted substantive terms and compound terms, frequency of occurrence and Web-IDF are calculated, after which, TF-Web-IDF weighting is performed.
5) The substantive terms and the compound terms are arranged in the order of their weighting, and the top 50 weights are acquired as the candidate answers.

Table 4 shows the candidate answers and weights obtained for the example shown in Fig. 2.

Table 4: Typical candidate answers acquisition.

Search term "Honshu-northernmost...town"	
Candidate answer	Weight
Aomori-Prefecture-Oma	699.298
Hokkaido-Shiokubi-Cape	39.582
Shimokita-Wine	13.194
….	….

6.3 Conceptualization

For acquisition of undefined term attributions, the search terms and the candidate answers are first conceptualized (Table 5) by the procedure described in Sec. 4.3. This enables the degree of association to be calculated using the answer scoring procedure described in Sec. 6.4.

The Web documents used for the search term conceptualization are acquired in the Web search, with the search term constructed from the question keywords as the input. This facilitates comprehensive processing of the documents needed for the query answer as well as the acquisition of their semantic features as attributes. In other words, the conceptualized search term can readily serve as the object of the query suitability assessment.

Table 5: Typical conceptualization (with partial list of attributes).

Search term: "Honshu-northernmost…town"		Candidate answer: "Aomori-Prefecture-Oma"	
Attribute	Weight	Attribute	Weight
Oma	1859.522	Oma	4035.559
Honshu	1522.526	map	188.496
town	276.172	weather	151.084

6.4 Answer scoring

The candidate answer is scored from the two perspectives of "suitability to the query" and "suitability to the answer type". The former is described in Sec. 6.4.1, and the latter in Sec. 6.4.2. The answer score calculation method is shown in Sec. 6.4.3.

6.4.1 Score calculation for suitability as answer to query

The degree of association $DoA(ac_i,QW)$ of candidate answer ac_i to search term QW (taken as the search term degree of association) is then calculated, and the result is taken as the score of the candidate answer suitability as an answer to the query.

6.4.2 Score calculation for suitability to answer type)

The node score $NodeValue(ac_i,AnswerType)$ of candidate answer ac_i relative to the answer type $AnswerType$ is calculated from Eq. (1). This calculation utilizes information held specifically by the answer type and, taken together with the degree of association between the candidate answer, the answer type and the node score, thus provides a more detailed quantification of the degree of association.

$$NodeValue(ac_i, AnswerType) = DoA(ac_i, AnswerType) \times \log(VerbHit(ac_i, AnswerType)) \quad (1)$$

$DoA(ac_i, AnswerType)$ represents the degree of association between the candidate answer and the answer type, while $VerbHit(ac_i,AnswerType)$ represents the number of hits obtained in a search using the search engine for a keyword linking a node verb to the candidate answer.

A node verb is itself a keyword defined by a relationship in which a specific term affiliated with a given answer type is the same as a verb that is accompanied by a particle that immediately follows that concrete term. For example, "祭"("festival") and "オリンピック" ("Olympics"), which may both belong to "event", may both be immediately followed by "に参加" ("participate" ("参加") "in" (postpositional particle "に")). By seeking Web hits using this category of association, it is possible to assess whether the candidate answer is suitable for the answer type. In this paper, node verbs are provided for all twelve of the abovementioned candidate-answer types (e.g., "を販売" ("sell" ("販売" and postpositional direct-object particle ("を") for "製品" ("product").

Utilization of the node score enables flexible numerical value calculation of the suitability of any kind of term, including proper nouns, to answer types.

6.4.3 Method of answer score calculation

The answer score $Score(ac_i)$ is calculated as follows:

$$Score(ac_i) = DoA(ac_i, QW) \times NodeValue(ac_i, AnswerType) \quad (2)$$

As shown, it is obtained as the product of the search term degree of association $DoA(ac_i,QW)$ and the node score $NodeValue(ac_i,AnswerType)$. The degree of association of the search term is an indication of whether the candidate answer is appropriate to the query, while the node score is an indication of whether the candidate answer is appropriate to the answer type. The answer score thus

combines both of these perspectives, and the higher its value, the more suitable the candidate answer is judged to be as an answer to the query. Table 6 shows the scores obtained for the example shown in Fig. 2.

Table 6: Typical scoring results.

Candidate answer	Search term degree of association	Node score	Answer score
Aomori-Prefecture-Omamachi	0.107	0.0575	0.0062
Hokkaido-Shiokubi-Cape	0.07	0.0071	0.0005
Shimokita-Wine	0.008	0.0004	0

7 Evaluation of system and components

To evaluate the method proposed in this paper, as embodied by the iWR system and two of its components, a test set of 120 questions having proper nouns as the correct answers was compiled from a questionnaire. For each of the questions in the test set, both the correct answer and the correct answer type were thus given, as shown in Table 7, and each correct answer type was one of the twelve candidate answer types shown in Table 1.

Table 7: Example of question used in the iWR system and components evaluation.

Question	Where is the northernmost town in Honshu?
Correct answer type	Place
Correct answer	Omamachi, Aomori Prefecture

Both the determination of the query answer type (Sec. 7.1) and the acquisition of the candidate answers (Sec. 7.2) were evaluated, as was the overall iWR system, including the answer scoring (Sec. 7.3).

7.1 Evaluation of answer type determination

In this study, the knowledge base for answer type determination had been constructed from the given correct answer types for all of the queries of the test set, and the answer type determination for the queries was therefore presumably invariably successful. This cannot be expected to apply, in general, for all queries. Therefore, it was necessary to investigate the success rate in response to input questions that are not so represented in the knowledge base, rather than one in which the knowledge base could be applied to all input questions. Accordingly, we assessed the accuracy of the query answer type determination without the provision of knowledge of the correct answers to the input questions of the test set.

The output answer types that matched the assumed correct answer type were deemed correct (○), those that did not match the assumed correct answer type but were correct in the output of "person" were deemed partially correct (Δ), and those that did not match the correct answer type were deemed incorrect (×).

The results were: 96 correct, 14 partially correct, and 10 incorrect. With both correct and partially correct deemed correct, the number of correctly determined problems was thus 110 (91.7%).

7.2 Evaluation of candidate answer acquisition

Candidate answer acquisition was evaluated by investigating whether correct answers could be acquired by performing acquisition of candidate answers for the 120 questions. Cases in which a correct answer was acquired were indicated by a circle (○) and those in which no correct answer was acquired were indicated by a cross out (×). Candidate answers containing part of the notation of the correct answer, and those containing variant notations, were both deemed correct.

The results showed the acquisition of 113 (94.1%) correct answers for the 120 questions. The results also showed an average acquisition of four correct answers per for each query.

7.3 Evaluation of complete system

7.3.1 Evaluation method and measures

Evaluation of the proposed method, as embodied in the iWR system, may reasonably be based on the two essential aspects of its performance: whether correct answers to questions can be acquired, and whether the correct answers to the questions are among the high ranking outputs. The following were therefore used in this evaluation as measures of its performance. Output answers containing part of the notation of the actual correct answer or notational variants were deemed to be correct answers.

1) Accuracy, which is defined in this evaluation as the proportion in which the correct answer is among the top five ranking outputs of the system.

2) Mean Reciprocal Rank (MRR)

MRR as used here provides a measure of the "solution precision" of the system. Reciprocal Rank (RR) is the reciprocal of the rank of the highest ranking output candidate answer at which the actual correct answer occurs in the candidate answer list output by the system. MRR is the average value for all queries. RR and MRR each take values between 0 and 1, with higher values indicating higher solution precision of the system. In this evaluation, if the correct answer to a question does not occur in the top five ranks, the retrieval is deemed a failure and the RR value is taken as 0.

7.3.2 Evaluation results and discussion

Figure 3 shows the distribution of the ranks at which the correct answers appeared in the list of scored candidate answers as output by the iWR system. The failures in Fig. 3 are cases in which the correct answer did not appear among the top five ranks. The correct answer occurred in the rank-one output answer for 51 of the queries, and among the top five ranks for 91 (75.8%) of the queries.

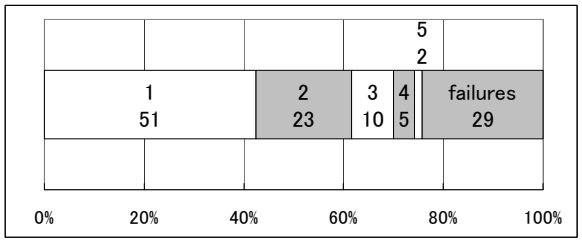


Figure 3: Distribution of correct answer occurrence in iWR system outputs.

Table 8: Results of evaluation.

Accuracy	MRR
0.758	0.562

Table 8 shows the system accuracy and MRR, as calculated from the evaluation results shown in Fig. 3.

For answers acquired as described in Sec. 6.4.1 (based on suitability as answer to query), the answer accuracy was 0.667 and the MRR was 0.470. For answers acquired as described in Sec. 6.4.2 (based on suitability to answer type), the answer accuracy was 0.5 and the MRR was 0.284. All of these evaluation results, taken together, clearly indicate that the proposed method, in which the answer score is obtained as the product of the score for suitability as an answer to the query and the score for suitability to answer type, can increase the accuracy of answer retrieval.

8 Conclusion

In this paper, we have proposed an intelligent retrieval method for selection of answers from the Web in response to questions. The system embodying this method incorporates a concept base that expresses the semantic features of terms in both words and weightings, together with components for calculation of degree of semantic association between terms as a numerical value, undefined term attributes acquisition using Web information that permits various expressions, optimum node determination, and question semantics understanding. As shown by the trial performance evaluation, the system enables flexible candidate answer scoring based on both its suitability to the query answer type and its suitability to the query. Thus, it can effectively retrieve answers in the form of terms being sought after by questions in interrogative sentence form.

Acknowledgements

This research has been partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (Young Scientists (B), 21700241).

References

[1] N. Okumura, E. Yoshimura, H. Watabe, and T. Kawaoka: "An Association Method Using Concept-Base", KES 2007/WIRN2007, Part I, LNAI4692, pp.604–611, 2007.

[2] H. Watabe and T. Kawaoka: "The Degree of Association between Concepts using the Chain of Concepts", Proc. of SMC2001, pp.877-881, 2001.

[3] H. Watabe, E. Yoshimura, and S. Tsuchiya: "Development of a System for Providing Current Events Information Based on User-Profile Information", Proc. of ICAI2010, Vol.I, pp.202-208, CSREA Press, 2010.

[4] K. Goto, N. Okumura, H. Watabe, and T. Kawaoka: "Understanding Support Method of Unknown Words Using Robot Type Search Engine", KES 2007/WIRN2007, Part I, LNAI4692, pp.631–638, 2007.

[5] G.Salton and C.Buckley: "Term-weighting approaches in automatic text retrieval", Information Processing and Management, Vol.41, No.4, pp. 513–523, 1988.

[6] NAIST Computational Linguistics Lab.: http://chasen-legacy.sourceforge.jp/

[7] Google: http://www.google.co.jp/

Analysing Expert System Mechanism

Sahil Gupta[1]
[1]Department of Information Technology, Maharaja Surajmal Institute Of Technology,
Guru Gobind Singh Indraprastha University, New Delhi, India

Abstract - *No matter which area of business one is engaged in, expert systems can be used to fulfill the need for higher productivity and reliability of decisions. Expert systems are most valuable to organizations that have a high-level of know-how experience and expertise that cannot be easily transferred to other members. They are designed to carry the intelligence and information found in the intellect of experts and provide this knowledge to other members of the organization for problem-solving purposes. An expert system is a set of programs (or a software) that manipulates encoded knowledge to solve problems or uncertainties in a specialized domain that normally requires human expertise. An expert system in essence captures the life-time experience of knowledge domain experts and reasons through this knowledge to solve problems. Today, when there is an increasing need for producing optimal and consistent solutions to problems in organizations, expert systems can make a significant contribution by making expertise available to the decision makers and technicians who need reliable answers quickly. This technical paper aims at providing an insight into the underlying technology of an expert system. Deployment of expert systems for financial analysis and air traffic management have been chosen to illustrate the techniques employed by an expert system in problem solving.*

Keywords: Expert Systems, Knowledge Domain Experts, Financial Analysis, Air Traffic Management

1 Introduction

Artificial intelligence programs that achieve expert-level competence in solving problems in specific domain areas by bringing together expert level knowledge in that domain are referred to as knowledge based or Expert systems. In a nutshell, expert systems solve problems that are normally solved by human experts. To solve expert-level problems, expert systems need access to a substantial domain knowledge base, which must be built as efficiently as possible. They also need to exploit one or more reasoning mechanisms to apply their knowledge to the problems they are given. Then they need a mechanism for explaining what they have done to the users who rely on them. Moreover, once the expert knowledge has been encoded in some form and loaded into a knowledge base, continuous efforts must be made to refine and improve upon this knowledge throughout the lifetime of the system.

2 Building blocks of an expert system

Successful expert systems will be those that combine facts and heuristics and thus merge human knowledge with computer power in solving problems. For this reason, expert systems are organized in three distinct levels:

2.1 Knowledge Base

The knowledge base constitutes the problem-solving rules, facts, or intuition that a human expert might use in solving problems in a given problem domain. The *knowledge base* of expert systems contains both factual and heuristic knowledge.

Factual knowledge is that knowledge of the task domain that is widely shared, typically found in textbooks or journals, and commonly agreed upon by those knowledgeable in the particular field.

Heuristic knowledge is the less rigorous, more experiential, more judgmental knowledge of performance. In contrast to factual knowledge, heuristic knowledge is rarely discussed, and is largely individualistic. It is the knowledge of good practice, good judgment, and plausible reasoning in the field. The knowledge base is usually stored by using a particular technique of knowledge representation.

Knowledge representation formalizes and organizes the knowledge. One widely used representation is the *production rule*, or simply *rule*. A rule consists of an IF part and a THEN part (also called a *condition* and an *action*). The IF part lists a set of conditions in some logical combination. The piece of knowledge represented by the production rule is relevant to the line of reasoning being developed if the IF part of the rule is satisfied; consequently, the THEN part can be concluded, or its problem-solving action can be taken. Expert systems whose knowledge is represented in rule form are called ***rule-based systems*.**

A knowledge base is created by ***knowledge engineers***, who translate the knowledge of real human experts into rules and strategies. These rules and strategies can change depending on the prevailing problem scenario. The knowledge base provides the expert system with the capability to recommend directions for user inquiry. A knowledge base is the nucleus of the expert system structure.

2.2 Working Memory

It refers to task-specific data for the problem under consideration. The working memory represents relevant data for the current problem being solved. The contents of the working memory, sometimes called the data structure, changes with each problem situation. Consequently, it is the most dynamic component of an expert system, assuming, of course, that it is kept current.

2.3 Inference Engine

The inference engine is the control mechanism that organizes the problem data and searches through the knowledge base for applicable rules. It applies the axiomatic knowledge in the knowledge base to the task-specific data to arrive at some solution or conclusion.

The ***problem-solving model*** used by inference engines organizes and controls the steps taken to solve the problem. One common but powerful model involves chaining of IF-THEN rules to form a line of reasoning. If the chaining starts from a set of conditions and moves toward some conclusion, the method is called ***forward chaining***. It is also known as data driven reasoning. If the conclusion is known (for example, a goal to be achieved) but the path to that conclusion is not known, then reasoning backwards is called for, and the method is ***backward chaining***. It is also known as goal driven reasoning. These problem-solving methods are built into program modules and are used by *inference engines* to manipulate and use knowledge in the knowledge base to form a line of reasoning.

3 Prominent Features of Expert System

Though an expert system consists primarily of a knowledge base, a working memory and an inference engine, some of other features are worth mentioning:

3.1 Domain Specificity

Expert systems are typically very domain specific. For example, a diagnostic expert system for troubleshooting computers must actually perform all the necessary data manipulation as a human expert would. The developer of such a system must limit his or her scope of the system to just what is needed to solve the target problem. Special tools or programming languages are often needed to accomplish the specific objectives of the system.

3.2 Heuristic Reasoning

Expert systems base their reasoning process on symbolic manipulation and heuristic inference procedures that closely match the human thinking process.

3.3 Explanation Facility in Expert Systems

One of the key characteristics of an expert system is the explanation facility. With this capability, an expert system can explain how it arrives at its conclusions. The user can ask questions dealing with the what, how, and why aspects of a problem. The expert system will then provide the user with a trace of the consultation process, pointing out the key reasoning paths followed during the consultation. Sometimes an expert system is required to solve other problems, possibly not directly related to the specific problem at hand, but whose solution will have an impact on the total problem-solving process. The explanation facility helps the expert system to clarify and justify why such a digression might be needed.

3.4 Data Uncertainties

Expert systems are capable of working with inexact data. An expert system allows the user to assign probabilities, certainty factors, or confidence levels to any or all input data. This feature closely represents how most problems are handled in the real world. An expert system can take all relevant factors into account and make a recommendation based on the best possible solution rather than the only exact solution.

3.5 Tools, Shells and Skeletons

Currently there are only a handful of ways in which to represent knowledge, or to make inferences, or to generate explanations. Thus, systems can be built that contain these useful methods without any domain-specific knowledge. Such systems are known as *skeletal systems, shells*, or simply *AI tools*. Building expert systems by using shells offers significant advantages. A system can be built to perform a unique task by entering into a shell all the necessary knowledge about a task domain. The inference engine that applies the knowledge to the task at hand is built into the shell. If the program is not very complicated and if an expert has had some training in the use of a shell, the expert can enter the knowledge himself. Many commercial shells are available today, ranging in size from shells on PCs, to shells on workstations, to shells on large mainframe computers. They range in price from hundreds to tens of thousands of dollars, and range in complexity from simple, forward-chained, rule-based systems requiring two days of training to those so complex that only highly trained knowledge engineers can use them to advantage.

3.6 Special Programming Languages

Expert systems are typically written in special programming languages. The use of languages like LISP and PROLOG in the development of an expert system simplifies the coding process. The major advantage of these languages, as compared to conventional programming languages, is the simplicity of the addition, elimination, or substitution of new rules and memory management capabilities. Some of the distinguishing characteristics of programming languages needed for expert systems work are:

- Efficient mix of integer and real variables
- Good memory-management procedures
- Extensive data-manipulation routines
- Incremental compilation
- Tagged memory architecture
- Optimization of the systems environment
- Efficient search procedures

3.7 User Interface

The initial development of an expert system is performed by the expert and the knowledge engineer. Unlike most conventional programs, in which only programmers can make program design decisions, the design of large expert systems is implemented through a team effort. A consideration of the needs of the end user is very important in designing the contents and user interface of expert systems.

4 Need for Expert Systems

Expert systems are necessitated by the limitations associated with conventional human decision-making processes coupled with the weaknesses inherent in conventional programming and traditional decision-support tools. Large Organizations do not rely on any single individual for performance but perform tasks or make decisions with the help of number of skilled employees who work together as a team to provide the necessary services. These members may belong to different sub-domains and thus expert systems are of vital importance in such cases as they may used to acquire the relevant knowledge and answers that are best suited for problem under consideration. Moreover, in this ever changing world scenario, an organization has to constantly upgrade and change it problem solving actions according to its needs. An efficient expert system is a reliable solution in such cases.

4.1 Expert System Vs Conventional Program

Applications that are computational or deterministic in nature are not good candidates for expert systems. Traditional decision support systems such as spreadsheets are very mechanistic in the way they solve problems. They operate under mathematical and Boolean operators in their execution and arrive at one and only one static solution for a given set of data. Calculation intensive applications with very exacting requirements are better handled by traditional decision support tools or conventional programming. The best application candidates for expert systems are those dealing with expert heuristics for solving problems. Conventional computer programs are based on factual knowledge, an indisputable strength of computers. Humans, by contrast, solve problems on the basis of a mixture of factual and heuristic knowledge. Heuristic knowledge, composed of intuition, judgment, and logical inferences, is an indisputable strength of humans. Therefore, problems that require heuristic reasoning are ideal candidates for expert systems.

Table 1. Comparison between Conventional Program and Expert System

Characteristic	*Conventional Program*	*Expert System*
Control by ...	Statement order	Inference engine
Control and data	Implicit integration	Explicit separation
Control Strength	Strong	Weak
Solution by ...	Algorithm	Rules and inference
Solution search	Small or none	Large
Problem solving	Algorithm is correct	Rules
Input	Assumed correct	Incomplete, incorrect
Unexpected input	Difficult to deal with	Very responsive
Output	Always correct	Varies with problem
Explanation	None	Usually
Applications	Numeric, file, and text	Symbolic reasoning
Execution	Generally sequential	Opportunistic rules
Program design	Structured design	Little or no structure
Modifiability	Difficult	Reasonable
Expansion	Done in major jumps	Incremental

4.2 Benefits of Expert Systems:

Expert systems:

1. Increase the probability, frequency, and consistency of making good decisions.
2. Help distribute human expertise.
3. Facilitate real-time, low-cost expert-level decisions by the non-expert.
4. Enhance the utilization of most of the available data.
5. Permit objectivity by weighing evidence without bias and without regard for the user's personal and emotional reactions.
6. Permit dynamism through modularity of structure.
7. Free up the mind and time of the human expert to enable him or her to concentrate on more creative activities.
8. Encourage investigations into the subtle areas of a problem.

5 Fineva- An Expert System for financial analysis of Firms

Special attention is being paid to create expert systems in the domain of finance, so that the domain knowledge could be accessible to a wider circle of people in the first place, and also to make the work in this field easier. One such expert system is Fineva or Financial Evaluation which can be used for financial analysis of firms.

The complete methodology for knowledge acquisition and representation in the field of financial analysis is implemented

in the system called FINEVA (FINancial EVAluation) (Matsatsinis, Doumpos & Zopounidis 1997). The
FINEVA system is a multicriteria knowledge-based decision support system for the assessment of corporate performance and viability. The system has been developed using the M4 expert system shell, by N.F. Matsatsinis, M. Doumpos and C. Zopounidis of Technical University on Crete.
Financial analysis of firms involves identification of the strengths and weaknesses of firms, mainly through
judgemental procedures concerning the qualitative evaluation and interpretation of financial ratios. The technology of
Expert systems (ES-s) technology is well suited to these kinds of tasks. The symbolic reasoning of ES-s enables them
not only to draw conclusions, through a process similar to the one used by human experts, but also to provide explanations concerning their estimations.
In the development of FINEVA, the knowledge from the international literature has been used and further knowledge acquisition has been conducted through a series of interviews with the financial experts from a bank in
Greece. Decision tables have been used to elicit the knowledge from the experts in the most efficient way, while a decision tree provides a graphical representation of the acquired knowledge. The representation through production rules is used to implement the acquired heuristics in a knowledge base.
The output that FINEVA produces is a specific ranking of the firms considered, according to a class of risk. The assessment of corporate performance and viability is achieved through the combination of the evaluation of financial status and the qualitative evaluation of the firm. For the two basic components the experts proposed equal weights.

5.1 Financial Ratios

5.1.1 Profitability Ratios:

The profitability is evaluated through the examination of the profitability of total assets (industrial profitability) and stockholder's equity (financial profitability), the gross profit to total assets ratio, and the profit margin.

5.2.2 Solvency Ratios:

The evaluation of solvency is accomplished by the analysis of the debt capacity (the short-term, the long-term and the global debt capacity) and the liquidity of the firm (the direct and the general liquidity)

5.2.3 Managerial Performance Ratios:

The expenses (interest expenses, general and administrative expenses) and the mean period between accounts receivable and accounts payable are combined to evaluate the managerial performance of a firm.

5.2 Qualitative Criteria

The qualitative criteria for qualitative evaluation of the firm are: manager's work experience, firm's market niche/position, technical structure-facilities, organization personnel, and market flexibility.

Each one of these criteria has been modeled using a five-point scale: not satisfactory, medium, satisfactory, very satisfactory and perfect. Table 2 and 3 show some examples of how the above financial ratios and the qualitative criteria are represented in FINEVA.

Table 2: Modeling of Financial Ratios in FINEVA

Industrial profitability A1: A1 < 10% not satisfactory 10% < A1 <= 20% medium 20% < A1 <= 30% satisfactory A1 > 30% very satisfactory	**Financial profitability** A2: A2 <= 17.5% not satisfactory 17.5% < A2 <= 20% medium 20% < A2 <= 23% satisfactory 23% < A2 very satisfactory
Gross profit/Total assets A3: A3 <= 0% not satisfactory 0% < A3 <= 50% medium 50% < A3 <= 75% satisfactory A3 > 75% very satisfactory	**Profit margin** A4: A4 <= 0% not satisfactory 0% < A4 <= 50% medium 50% < A4 <= 100% satisfactory A4 > 100% very satisfactory
Short-term debt capacity B1: B1 < 25% not satisfactory 25% < B1<= 50% medium 50% < B1 <= 75% satisfactory 75% < B1 <= 100% very satisfactory	**Global debt capacity** B2: B2 > 80 % not satisfactory 60% < B2 <= 80% medium 40% < B2 <= 60% satisfactory B2 <= 40% very satisfactory
Long-term debt capacity B3: B3 <= 0.5 satisfactory B3 > 0.5 not satisfactory	**General liquidity** B4: B4 >= 2 satisfactory B4 < 2 not satisfactory
Direct liquidity B5: B5 <= 1 not satisfactory 1 < B5 < 1.5 satisfactory B5 >= 1.5 very satisfactory	**Financial expenses** C1: C1 > 5% not satisfactory 3% < C1 <= 5% medium 2% < C1 <= 3% satisfactory C1 <= 2% very satisfactory
General and administrative expenses C2: C2 > 8% not satisfactory 6% < C2 <= 8% medium 4% < C2 <= 6% satisfactory 2% <C2 <= 4% very satisfactory C2 <= 2% perfect	**Received period of accounts receivable** C3 > C4 not satisfactory C3 <= C4 satisfactory
Circulation of inventories C5: C5 increasing not satisfactory C5 reducing or stable satisfactory	**Circulation of customers and notes receivable** C6 <= C7 satisfactory C6 > C7 not satisfactory

Table 3: Modeling of Qualitative criteria in FINEVA

Managers' work experience:	
Negative experience	Not satisfactory
No experience	Medium
Positive experience up to 5 years	Satisfactory
Positive experience 5-10 years	Very satisfactory
Positive experience more than 10 years	Perfect
Firms' market niche/position:	
Strong competition, firm's weak position	Not satisfactory
Strong competition, established and competitive firm	Medium
Moderate competition, firm's strong position	Satisfactory
Weak competition, firm's leadership position	Very satisfactory
Single position, monopoly	Perfect
Technical structure-facilities:	
Old and inappropriate equipment, outdated production methods	Not satisfactory
Moderate technical structure, non-competitive production cost	Medium
Relatively modernized equipment	Satisfactory
Sound technical structure, full modernization scheme under way	Very satisfactory
Excellent structure, modern production Methods	Perfect
Organization personnel:	
Lack of organization/staff hiring policy	Not satisfactory
Moderate organization/staff hiring policy	Medium
Moderate organization/staff hiring policy, willingness to improve	Satisfactory
Good organization/staff hiring policy	Very satisfactory
Excellent organization/staff hiring policy	Perfect
Market flexibility:	
The firm does not follow market trends, produces low-demand products	Not satisfactory
The firm has a limited flexibility	Medium
The firm has a satisfactory flexibility	Satisfactory
The firm follows market trends	Very satisfactory
The firm is a leader in its production branch activity	Perfect

Table 4: Some rules for FINEVA Knowledge Base

IF Direct-liquidity = satisfactory **AND** General-liquidity = not-satisfactory **THEN** liquidity = medium	**IF** Financial-status = very-satisfactory **AND** Qualitative-evaluation = satisfactory **OR** Qualitative-evaluation = very-satisfactory **OR** Qualitative-evaluation = perfect **THEN** Expert-system-evaluation = very-satisfactory

Along with the production rules, some meta-rules were also necessary to represent the heuristics that the experts use in several practical cases when assessing the performance of a firm, as well as to reduce the time needed to reach the final estimation.

In cases when the value of a criterion is unknown, the limited information is ignored by the system by directing the inference process to the corresponding set of rules which do not examine the unknown information. The accuracy of

the conclusion reached depends on the amount of the available data:

When cached **(financial-expenses is unknown or general-and-administrative-expenses is unknown) = rule-1630.**

FINEVA's inference engine that draws conclusions about the performance of the examined companies employs both the forward- and the backward-chaining method. The forward-chaining method is used to guide the inference process to a set of rules (meta-reasoning), and the backward-chaining method is applied within this set to derive a conclusion. This inference strategy closely reflects human-expert logic and decision making in the domain.

6 Air Traffic Management/Control

Air traffic control (ATC) is a service provided by ground-based controllers who direct aircraft on the ground and in the air. The primary purpose of ATC systems worldwide is to separate aircraft to prevent collisions, to provide correct trajectories and exact location coordinates for arrival and departure of aircrafts taking into account multiple factors, to organize and expedite the flow of traffic, and to provide information and other support for pilots when able. In some countries, ATC may also play a security or defense role (as in the United States), or be run entirely by the military (as in Brazil).In many countries, ATC services are provided throughout the majority of airspace, and its services are available to all users (private, military, and commercial). Depending on the type of flight and the class of airspace, ATC may issue instructions that pilots are required to follow, or merely flight information to assist pilots operating in the airspace. In all cases, however, the pilot in command has final responsibility for the safety of the flight, and may deviate from ATC instructions in an emergency. Efforts are being made to develop expert systems that can handle air traffic management and great progress has already been made in this area. One of the applications of expert systems in air traffic control being currently used is in managing air traffic through principled negotiation. The worldwide aircraft/airspace system (AAS) is faced with a large increase in air traffic in the coming decades, yet many flights already experience delays. The AAS is comprised of many different *agents*, such as aircrafts, airlines, and traffic control units.

Principled Negotiation between agents allows all the agents in the system to benefit from multiple independent declarative analyses of the same situation. It motivates an iterative optimization method. An initial master plan is formulated, specifying action for each agent. Agents repeatedly search for options (i.e. alternative master plans) that provide mutual gain.

When an agent finds an option that better meets its interest, it proposes the option. The other agents evaluate the option and accept or reject it. If there are no rejections, the option is implemented. If an agent rejects the option, it sends a message to the proposer explaining the reasons for rejection. The proposing agent uses this additional information to improve its search. The expert system essentially acts as a coordinator among different agents. Intelligent agents are modeled as rule based expert systems. Applied to air traffic operations, principled negotiation allows much greater freedom for optimization by system users while maintaining safety.

6.1 Runaway Slot Negotiation Example:

Expert systems based on principled negotiation can be an effective optimization method in situations where the agent actions are closely coupled, such as arrival sequencing at a busy airport. The acceptance rate of an airport (the number of available slots per hour) depend on the runaway configuration, the number of departures and the weather conditions. The Federal Aviation Administration (FAA) monitors the national AAS in United States of America. If the FAA predicts that arrivals will exceed the arrival rate of an airport, flow control restrictions are placed on aircraft departing for that airport. If a ground delay program is run departing aircrafts are given the time at which they can expect to receive clearance to depart, calculated from their allocated arrival slot time at the constrained airport. Currently airlines must accept the arrival slots they are allocated, unless they cancel a flight. If they cancel a flight, they can substitute another flight into this slot, thus freeing another slot. No swapping of flights between airlines is allowed. The airlines incur a number of costs from change in arrival time (e.g. extra crew duty time) which they want to minimize. The use of expert system can prove to be highly beneficial in such problem scenarios as it can provide reliable solutions quickly and also explain its choice for that particular solution.

Figure 1 explains the system used in air traffic management. It is a rule based system which employs the technique for principled negotiation.

7 Other Applications

Expert systems are designed and created to facilitate tasks in the fields of accounting, medicine, process control, financial service, production, human resources. Expert systems are now able to use innovative methods to provide viable solutions to complex problems in everyday applications. They are being used in specialized areas of high level control and reconfiguration logic. Intelligent aircraft systems such as drones used in military operations are also a product of expert systems. They have the ability to replace any task that would normally require a high level of human expertise to be done.

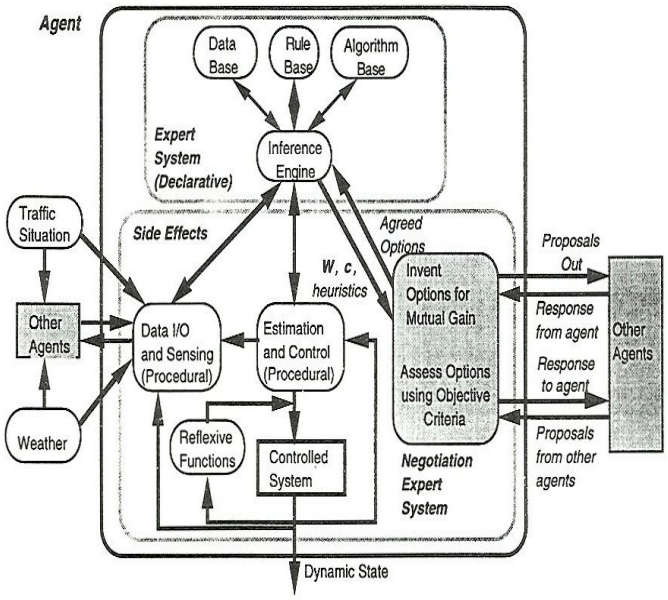


Figure 1: Air Traffic Management using Principled Negotiation

8 Conclusion

Expert systems can be thought of as computer simulations of human experts. They use knowledge rather than data to control the solution process. The knowledge is maintained in an entity separate from the control program which permits refinement of knowledge base without recompilation of control programs. Thus, the knowledge base in an expert system always grows and this maintains the competency of the expert system. They have the ability to reason through knowledge, handle uncertain data, provide an explanation for their choice of a particular solution. They are important tools which can be used to produce reliable solutions to complex problems which cannot be handled by conventional programs.

9 References

[1] Dan W. Patterson" Introduction to Artificial Intelligence And Expert Systems", Prentice Hall Inc. USA (Pages 326-357).

[2]. Robert S. Engelmore, Edward Feigenbaum "Expert Systems and Artificial Intelligence".

[3]. Ljubica Nedović , Vladan Devedžić"Expert Systems In Finance- A Crossection of Field" available online since 15 March 2002.

[**4].** John P. Wangermann and Robert F. Stengel"Principal negotiation between intelligent agents:a model for air traffic management" available online.

[5]."Fundamentals of Expert Systems",Wiley Inc.

A Web-based Prototype of Fish Image Searching System

Xitao Zheng, Yongwei Zhang
College of Information Technology, Shanghai Ocean University
LinGang New City, Shanghai, 201306 China
xtzheng@shou.edu.cn zhangyongwei_108@163.com

Abstract *- To develop a system that can identify a fish by its image, this paper summarizes the system which is being developed by our group to setup the fish geometric ontology base, convert image data to morphological features and compare these features with the ontology base. The positive result will return not only the fish name, it will also return the type it belongs to, and the intuit relation of the fish identified and provide links the existing web resources and display the interested properties of habit that a user may be interested, like the akin of the fish, the living places, and its food preferences. The system is proposed for both biological researcher and normal interested web users. The program is programmed in Java. The current knowledge bases are mostly in owl format and can be converted to oracle or Mysql for uniformed web application purposes.*

Keywords: Fish Searching, Ontology Base, Morphological Ontology, OWL

1 Introduction

While we are using internet to search information every day, we need to either know the website addresses that we are planning to go or know the key words that we know the interests belongs to. If we have no knowledge of the target we are holding, it will be very hard to get any information from the internet. This is because that nearly all information and communication on the web are based on strings or characters which have been virtualized and interpreted by human beings. Only by reading through a document or the result of a search, can people understand if this information is relevant to them or not. So you have a fish in hand, and have no idea of its name and no people around you know the fish, you probably will not be able to get anything from the internet by just searching a fish, a big fish, a small fish, a red fish, or a white fish[3]. These kinds of difficulties will arouse us to build some knowledge base that can store the information of a fish's geometric and color features and the applications that can disseminate the picture of a fish into these kinds of features.

There are some works that can prescreen fish by its roundness or flatness, its sizes or its entropy, from a given set, but none of them can be used to precisely identify or classify fishes. There are a lot of jobs and applications that use ontology to do professional search, i.e., the search that have relatively fixed properties and relationship, but none of the works are available for fish. This is partly because of the lack of proper definition of ontology that can be processed by computer. There are people that propose to use chemical, molecular, or genetic method to define the ontology and thus the classification; none of these kinds of jobs are successful, even manually. The most popular method is to identify the fish by its shape, and expert will also examine the fish details like fin and inner organ to determine an unknown species. So morphological features are still the dominant features for fish classification.

Plenty of works done recently to absorb the development of computer technology to build up agricultural knowledge base and therefore the reasoning method, but most of these kinds of job are still limited to the input of existing features that are described in the textbook and index them with the ontology properties. The searching and matching are still based on the test searching techniques. This kind of manual way can hardly be utilized by computer, because of the description can be random and test search is another uncertain thing to be researched. With all those works on OWL and fish OWL, the bottleneck is still the automatic identification of fish, without know what fish it is, there is no way to go the second step.

Our previous job [1] is trying to provide a way to break this barrier. A complicated geometric and color features set is defined to simulate the human view perceptions of a fish, and further works is done [2] to use these features to do matching and pattern recognition, which paved the road of a possible fish image searching system, or, more plainly speaking, the intelligent fish recognition system that can be accessed by public over the internet.
Ontologies are useful if better scoped and-defined for a specific domain and application. Processing techniques can be used to enrich ontologies, which in turn can be used for knowledge discovery and extraction. The system will be running on the server so the complicated recognition algorithm will be not be a problem when retrieving most of fish features from images, the natural links that can be expressed in ontology relationship can utilized the flexibility of internet to access any information that a user may interest related to the current fish, and thus verify if the system provides a good solution or it just did the wrong job.

For the fish image recognition, there are some papers that use pattern recognition to do rough classification, these kinds

of works are: fish identification from a given set, fish counting, fish rough counting using digital sonar image, ocean color analysis to find fishing area by the satellite images, etc[3][4]. There is no new definition of fish ontology so it can be used for pattern recognition and classification system. What is more, these are all proprietary systems and designed for special purposes, none of these applications is designed for public use and scientific educations [6].

An example [5] is that the three research results of the AOS (Agricultural Ontology Services) project in FAO (Food and Agriculture Organization) is constructing a primitive ontology in fishery area. The ontology consults ASFA (Thesaurus Aquatic Sciences Thesaurus), ARGOVOC thesaurus with multiple languages, that exposes the problem that how week a system is if it is a language relevant system.

2 Our Method

As we can see from the above information, we will build a language independent, human independent system that can be even used by a kid. This ideal situation: by flashing a fish in front of a camera, the computer will catch up a few pictures and send them to the remote server for recognition. The server will try to recognize it and prompting the kid to turn the fish around, places it closer or farther, or do it in the better illuminated environment. If the system does not find a match, it will keep sending instruction until the options are exhausted. In most cases, we can expect the system will return the name, habit and relationship information from the internet and the image withdrew from the internet should be the same or similar to the one that user is holding, which will accomplish a typical service action.

To accomplish this purpose, we need the system architecture as Fig1, where the user equipped with personal computer, internet access and camera or a cameraed fish image will pass through the image through the internet to the server, mapping into geometric ontology base, which is a few owl files, invoke the reason machine, and query the related knowledge base, which will relate to more related databases and spread the relation over the internet resources to get the total view of the species that we are working on. So a cross linked, inter-referenced knowledge reasoning machine will be the best reason for using ontology base for this kinds of interpretation.

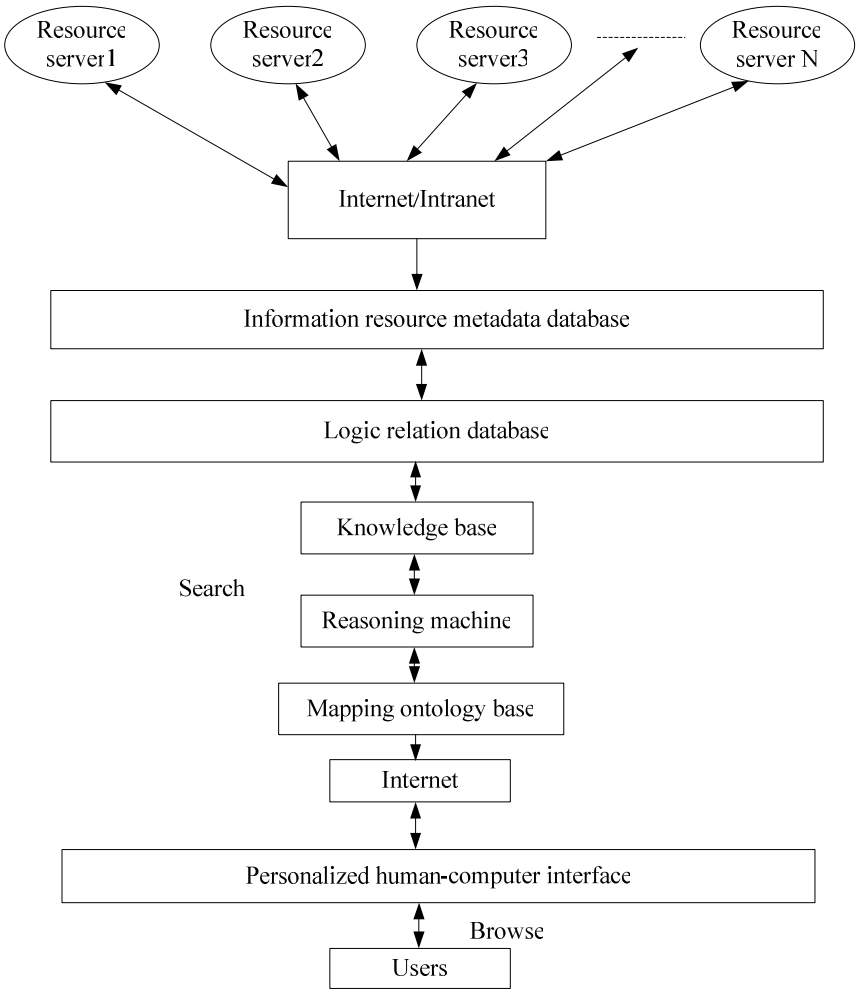


Fig1. Architecture of Fish Ontology Query Program

As a prototype, the no-matching return instruction system will not appear in this paper, only a straight functioning route is described here in Fig 2.

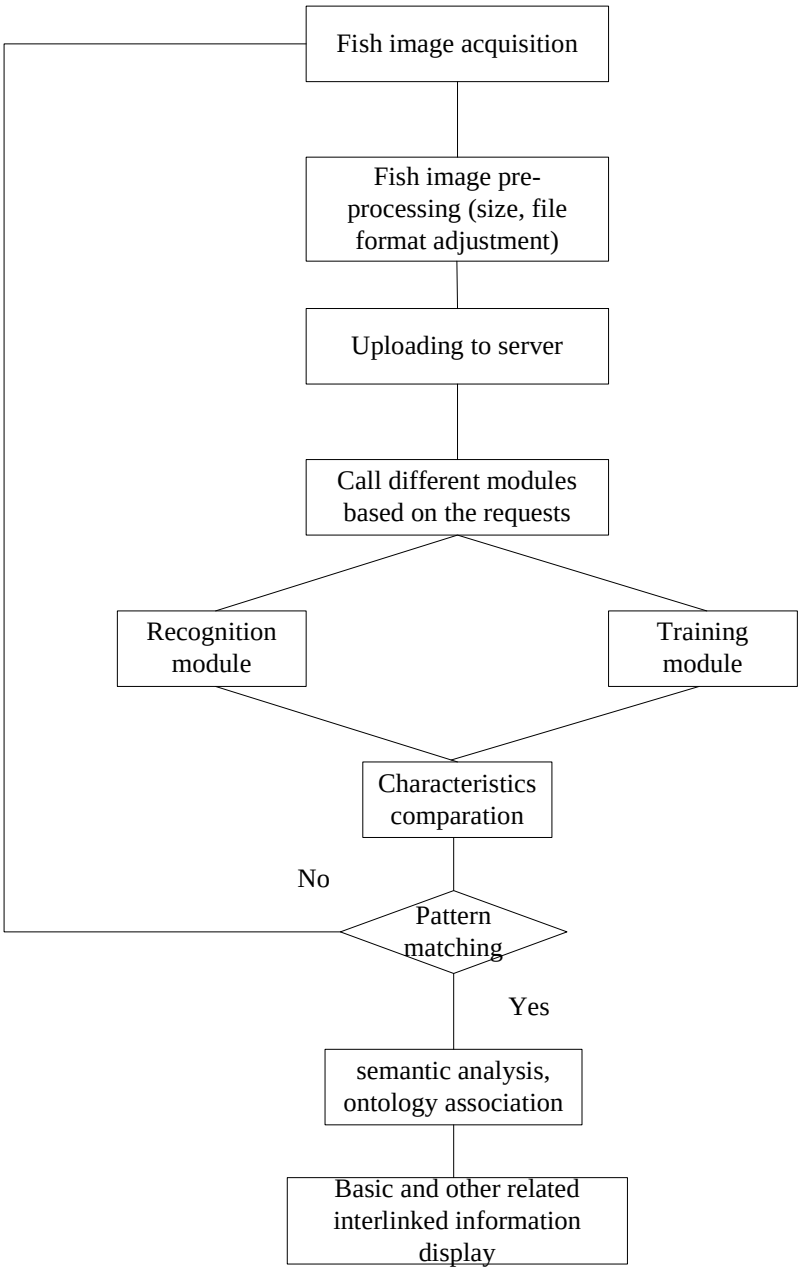


Fig2 Data flow chart of system

Fig 2 shows the data flow chart of the realization program across the platform.

In the actual application, the picture quality, the picture format and the shooting angles are all the problems that will lead to a failure search, so the future enrichment will be intelligent instruction system that can be based on the quality of the picture and the feed back to regenerate the instruction database.

The other problem is the feature definitions. Because of the huge amount of information caught by human eyes, the dozens of morphological features and color features will not be adequate to cover a fish in real life. Generally, more detailed features will help the recognition rate. In actual case, this may also decrease the recognition rate. This is because the cameraing environment can be random and the picture quality can be very different, so the features acquisition of the learning process may be a special case so the recognition picture will be discriminated.

3 Experiment

The experiment is done with a silver carp image as the target, which is shown in Fig 3. A few silver carps, common carps, snakeheads, mugil celphalus, which are shown in Fig 4 and Fig 5, are input as the learning process with word instructions.

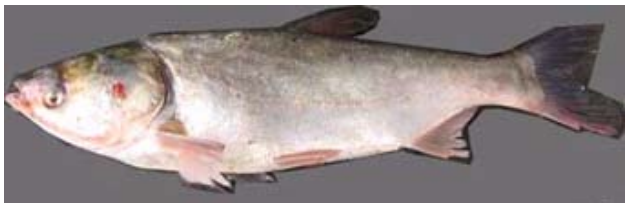


Fig.3 the Target image of silver carp

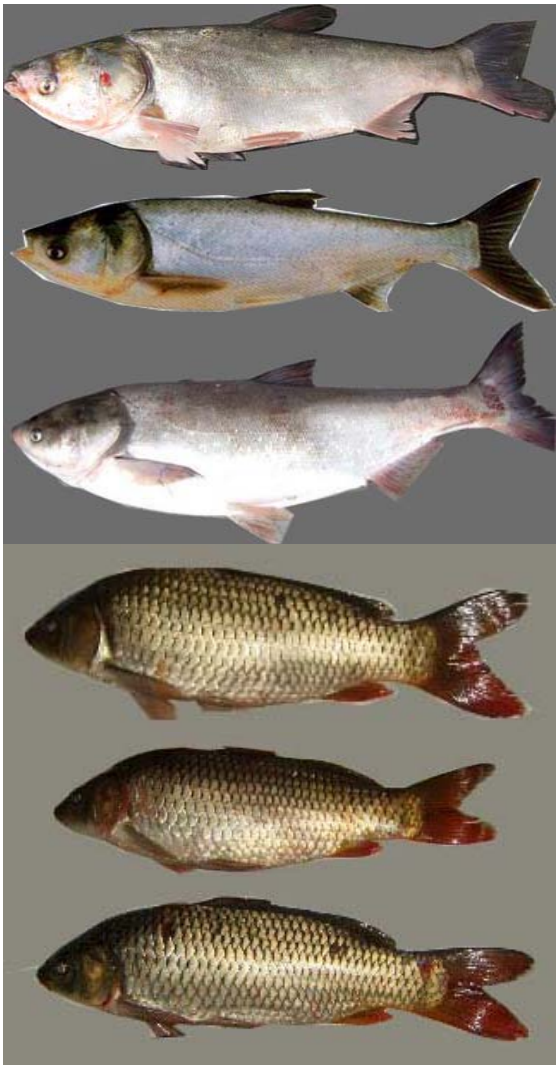


Fig.4 the Input images of silver carp and common carp

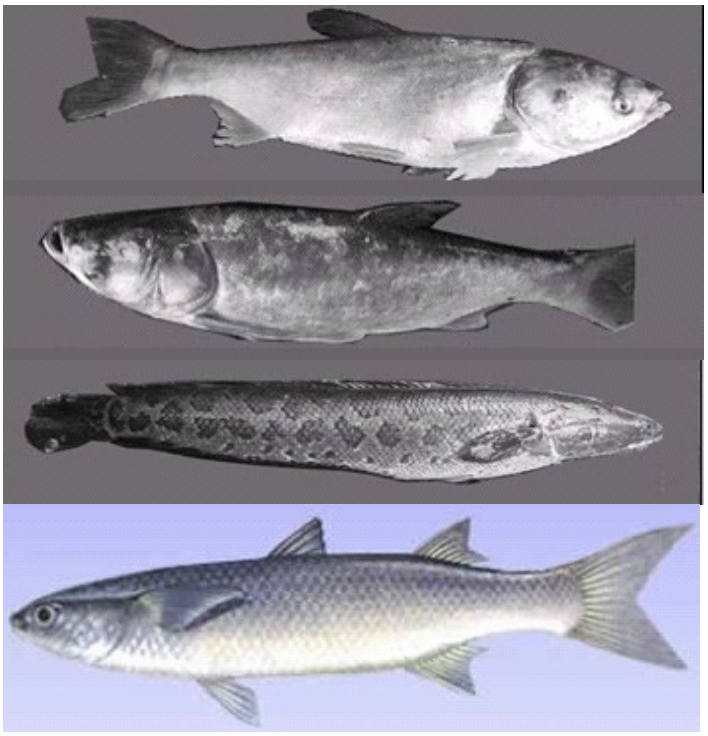


Fig.5 the Input images of more specimens of silver carp, the snakehead and the mugil celphalus

The recognition is instant and the information of silver carp can be shown in Fig 6. The system is in Chinese so the characters are all in Chinese.

If a picture of a crab or other fish which is not learned is input as the target image, the system will reject it immediately.

Fig 6 the system layout information of silver carp image search

4 Conclusion

This prototype described in this paper can successfully learn a fish's features from this picture, store these features in the web knowledge base. The system can also disseminate a target fish picture into pre-designed features and process the searching and reasoning action over the web, correct information will be returned in normal picturing condition.

The prototype is insufficient to provide service now because of the diversity of geometric definition and the huge variety of the fishes. More important, the picture quality and the shooting angle will limit the success retrieval of fish properties. It is expected an intelligent instruction system can be developed to instruct the user to use correct equipment (camera), shoot in correct angle, etc. So further research on ontology definition and further feature learning work need to be done in order to get a practical service system.

References

[1] Xitao Zheng, Yongwei Zhang. A New Morphological Ontology Method for Carp Based on Digital Image Measurement. Web Information Systems and Mining, 2010, page23-27.

[2] Xitao Zheng, Yongwei Zhang. An Ontology Method for Silver Carp Auto-Recognization Based on Digital Image. Web Information Systems and Mining, 2010, page331-334.

[3] C. Costa, A. Loy, S. Cataudella, D. Davis, M. Scardi. Extracting fish size using dual underwater cameras. Aquacultural Engineering, Volume 35, Issue 3, October 2006, Pages 218-227.

[4] Frank Storbeck, Berent Daan. Fish species recognition using computer vision. Fisheries Research, Volume 51, Issue 1, April 2001, Pages 11-15

[5] Qian ping, Zheng Yelu. The Research and Application of Agricultural Ontology. China agricultural science and technology press, 2006, Pages 51-52.

[6] Pan Hongqiang. The new practical technology of uncontaminated freshwater aquaculture. China agricultural science and technology press. 2005.

Tracking Evolutionary Links Among Coronavirus Types Using Self-Organizing Neural Networks

Francis Thamburaj[1], **and Gopinath Ganapathy**[2]
[1]Computer Science Department, Bharathidasan University, Tiruchirappalli, Tamil Nadu, India
[2]Computer Science Department, Bharathidasan University, Tiruchirappalli, Tamil Nadu, India

Abstract - *The Coronaviruses are worldwide in distribution and caused an epidemic in China in 2003. The rapidly mutating Coronaviruses spreads fast by taking different forms and infects not only human beings but also cattle, pigs, rodents, cats, dogs and birds. In fighting against these viruses, it is important to elucidate the evolutionary links among the different types of Coronaviruses. This article explores the possible roots of the evolution of Coronaviruses with the help of unsupervised Self-Organizing Map (SOM) neural network. The migration paths of these viruses are analyzed with different Self-Organizing maps based on the different genomic signatures of 50 complete Coronavirus genomes. The results are corroborated with other findings and thus Self-Organizing maps are proved to be fast, efficient, and economical tool to use as initial pointers to detailed phylogenetic analysis. New subgroups are also revealed by clustering these genomes with SOM.*

Keywords: Self-Organizing Feature Map, Coronavirus, Genome Classification, Nucleotide Signatures, Pattern Recognition, Abinitio Method

1 Introduction

The Coronaviruses were first isolated from chickens in 1937. The Coronaviruses belong to Coronaviridae family that has two genera Coronavirus and Torovirus. There are now approximately 15 species in this family. It infects not only man but cattle, pigs, rodents, cats, dogs and birds. Some are serious veterinary pathogens, especially chickens. In humans, they cause respiratory infections (common) including Severe Acute Respiratory Syndrome (SARS), enteric infections in infants less than 12 months (occasional) and neurological syndromes (rare). In February 2003, a severe acute respiratory syndrome coronavirus (SARS-CoV) emerged in humans in Guangdong Province, China, and caused an epidemic that had severe impact on public health, travel, and economic trade. Coronaviruses are worldwide in distribution, highly infectious, and extremely difficult to control because they have extensive genetic diversity, a short generation time, and a high mutation rate. Coronaviruses have been shown, both experimentally and in nature, to undergo genetic mutations and recombination at a rate similar to that of influenza viruses [1, 2].

All coronaviruses employ a common genome organization. The Coronaviruses (CoVs) are enveloped, plus- strand RNA viruses. The genomic RNA is 27 – 32 Kb in size, capped and polyadenylated. The virions are 80 – 150 nm in diameter and have a unique morphology, with extended, petal-shaped spikes which give the virus a crown-like projection. There are about 50 complete Coronavirus genomes are available in NCBI.

CoVs are classified into three groups based on phylogenetic and serological relationships. Group 1 and 2 consist of different mammalian coronaviruses, whereas bird viruses dominate group 3 [3]. The group 1 coronaviruses can be divided into the two genetic subgroups 1a and 1b [4]. Members of group 1a include canine coronavirus, FIPV, transmissible gastroenteritis virus (TGEV), and ferret enteric coronavirus. Group 1b includes porcine epidemic diarrhea virus (PEDV), human coronavirus NL63 (HCoV-NL63) and human coronavirus 229E (HCoV-229E) [5]. The human Coronavirus NL63 is a clinically important virus. The human coronaviruses (HCoV) can cause both lower and upper respiratory tract infections that range in seriousness from common colds to severe acute respiratory syndrome [6]. It can be associated with extra-respiratory manifestations, including central nervous system involvement. Like influenza, coronaviruses can be neurotropic and may lead to serious mental disorders like Schizophrenia [7].

Based on phylogenetic analysis designed to elucidate evolutionary links among viruses, SARS-CoV is believed to have branched from the modern Group 2 coronaviruses, suggesting that it evolved relatively rapidly. This is significant because SARS-CoV is likely still circulating in an animal reservoir (or reservoirs) and has the potential to quickly emerge and cause a new epidemic [1]. Unlike the other RNA viruses, HCoVs can easily mutate and recombine when different strains infect the same cells and give rise to a novel Coronavirus with unpredictable host ranges and pathogenicity [8].

The classification or grouping of different coronaviruses is important because that could provide useful insights for understanding the adaptation of animal coronaviruses to humans in general and the emergence of new coronaviruses. Although, the phylogenetic analysis techniques can give accurate lineages of different coronaviruses, it raises confusions (sometimes) that could be attributed to technical aspects, like the type of alignment algorithm that is employed, alignment quality, choice of viruses and genomes analyzed,

and software used. The nucleotide signature based clustering could give a valuable clue regarding the evolutionary pathway to human coronavirus, especially SARS-CoV. The clusters can be used as general pointers in the phylogeny of coronavirus genomes that can facilitate the development of adequate preventive and therapeutic measures. This article aims to provide such clues with the use of SOM as a tool.

2 Review of Literature

The Self-Organizing Map (SOM) is an unsupervised artificial neural network, unlike the supervised neural networks that has to be taught with sample data, which can cluster the given data [9]. It uses an unsupervised method for forming topologically correct representation of the data [10]. Topologically correct implies that the similar data are mapped onto the neighboring neurons. It consists of local data models located at the nodes of the low-dimensional map grid. In our experiment the low-dimension is fixed as two dimensions. The models are adapted via competitive learning process, where the neurons will compete with each other to become the winner. The winner will be the neuron that has closest neuronal weights to that of the input data.

There are two ways of constructing the SOM for processing the biosequence data processing. One way is to modify the SOM algorithm itself in order to fit the type of data at hand. The SOM for symbol strings could be used, as the biosequence data are made up of symbols (A, C, G, T). In 1996, Kohonen showed for the first time that SOM can be constructed for any data set for which a similarity or dissimilarity measure between its elements is defined [11]. This was based on the use of the batch-SOM training and the definition of the generalized median as the model associated with each SOM node. The first application of this method was the construction of SOM for symbol strings [12]. Two types of medians were used: the set median, which is an existing element of the data set, and the median, which does not have to be an exact replica of any element in the data set. The set median is applicable to dense data sets, while the latter kind of median may interpolate sparse data better [13].

Another possibility of constructing SOM is to convert the data into feature vectors so that the comparison and averaging can be done using familiar Euclidean distance and arithmetic averaging. Using the feature vectors of biosequences, SOM was used to extract uncommon sequences [14], to categorize interspecies genes [15], to analyze codon usage [16], to classify prokaryotic and eukaryotic genomes [17, 18], to identify atypical sequence composition [19], to motif discovery [20], to discover new groups in human endogenous retroviral sequences [21].

In this paper, 50 genomes of the Corona Virus are clustered using the unsupervised SOM based on the different feature vectors extracted from them. The aim of the paper is to understand the relationship and proximity of different Coronavirus types and hence to see the possible linkages among them.

The rest of the paper is organized as follows. The third part gives a short review of the SOM algorithm and fourth part explains the details of the experiment. The fifth part discusses various results of the classified maps. The sixth part summarizes the new findings of the experiments followed by the conclusion.

3 Review of SOM Algorithm

The SOM neural network consists of n x m neurons located at a regular low-dimensional grid, usually a 2-D n x m rectangular grid. The basic SOM algorithm is iterative. Each neuron i has a d-dimensional weight vector

$$w_i = (w_{i1}, w_{i2}, \dots, w_{id}), \qquad i = 1,2, \dots, nm$$

The initial values of all the weight vectors are given over the input space at random. The range of the elements of d-dimensional input data

$$x_j = (x_{j1}, x_{j2}, \dots, x_{jd}), \qquad j = 1,2, \dots, N$$

are assumed to be from 0 to 1.

The Algorithm

Step.1 Initialize all the models w_i's associated with the neurons of an m x n map.

Step.2 Randomly select an input vector $x(t) \in \Re^d$ from the training data set and is inputted to all the neurons at the same time in parallel.

Step.3 Distances between x_j and all the weight vectors are calculated using the Euclidean distance formula

$$d_j = \|W_j - X\| = \left(\sqrt{\sum_{j=1}^{nm} (w_{ij} - x_j)^2} \right)$$

where i = 1,2, ... , nm and j = 1,2, ... , N
A winner, denoted by c, is the neuron with the weight vector closest to the input vector x_j is found by

$$c = \arg \min_{1 \le i \le mn} \{ \| w_i(t) - x_j(t) \| \}$$

where ||.|| is the distance measure, Euclidean distance, x(t) is the input vector and $w_i(t)$ the model of neuron i at iteration t.
.
Step.4 The weight vectors of the neurons are updated as

$$w_i(t+1) = w_i(t) + \alpha(t)\, h_{cj}(t) \left[x_j - w_i(t) \right]$$

where t is the learning step; c is the index of the BMU; $h_{c,i}$ is called the neighborhood function.

Step.5 Steps 2 to 4 are repeated for each input vector until all w_i's converge to stable positions.

4 SOM Classification

The complete genomes of 50 Coronaviruses were collected from the NCBI website. The set includes the Avian, Murine, Bovine, Bat, Human, Porcine, SARS, Feline, Equine, Whale, Turkey, Thrush Coronaviruses. The feature extraction was done based on the different genome signatures. The feature vectors of monomers with 4 dimensions, dimers with 16 dimensions, codons with 64 dimensions and the RSCU value with 59 dimensions are constructed.

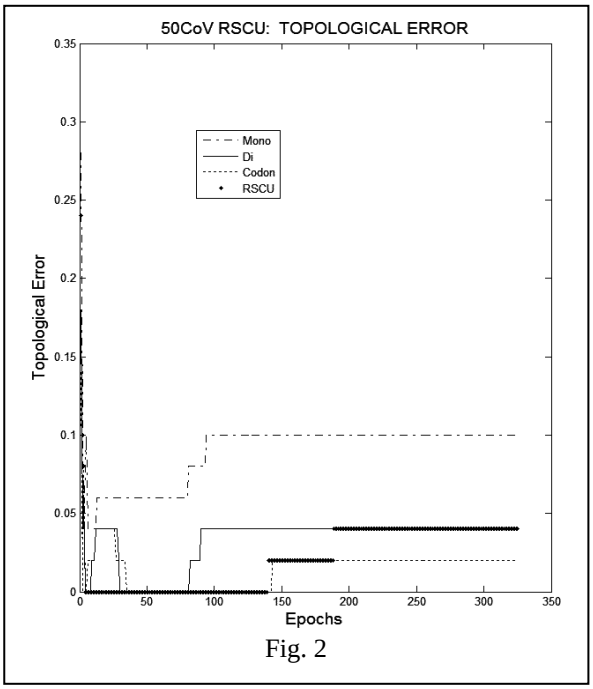


Fig. 2

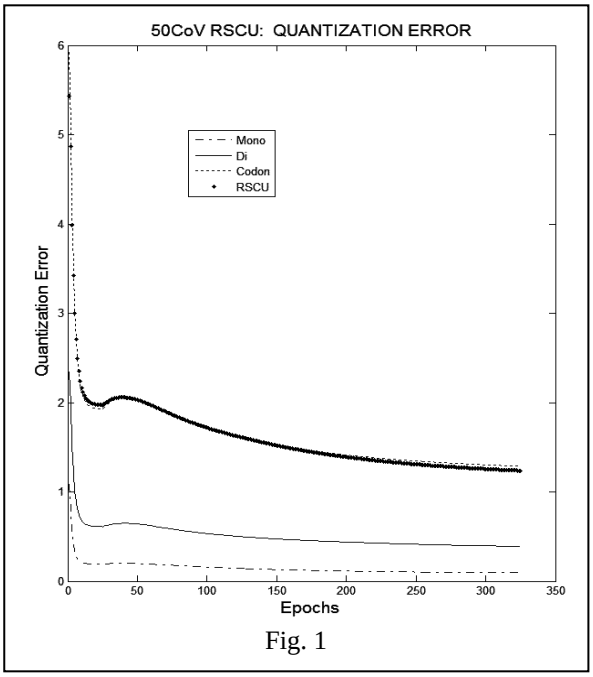


Fig. 1

The hexagonal 10x10 self-organizing neural network was linearly initialized based on the greatest eigenvectors of the data set. The Gaussian neighborhood is assumed. Monotonically decreasing learning rate followed with 0.7 as initial and 0.001 as the final rate of learning. Then the feature vectors are fed and resulting topological map was constructed after the fine tuning of the each stable map. Four maps with labeled grid were produced for monomer, dimer, codons and RSCU respectively (Fig.3, Fig.4, Fig.5, Fig.6). In each epoch, the Euclidean distance was used to find out the winning neuron.

Two quality measures of the four SOM maps are calculated and plotted for the comparative analysis. The first measure is the quantization error (Qe). It is calculated for each epoch for each map and the consolidated result was drawn against the number of epochs (Fig.1). The quantization error calculates the average distance between each input vector and its 'best matching unit' (BMU) [9]. Therefore, the small value of Qe is more desirable.

The quantization errors for codons and RSCU are slightly more than one and for the monomers and dimers they are less than 0.5. The least quantization error is achieved for monomers. The fine tuning of the maps have considerably reduced the Qe for the codons and RSCU maps.

Another quality measure of the SOM map is the topographic error (Te) (Fig. 2). It describes how well the SOM preserves the topology of the studied data set [22]. It calculates the average ratio of the number of nodes for which the first and the second winners are not neighbors to each other. Therefore small value of Te is more desirable. Unlike the quantization error it considers the structure of the map.

Here, the codon map has reached the minimum level followed by the RSCU and dimer maps. The monomer map has somewhat higher error rate but only 0.1 which is acceptable. All the other maps have less than 0.05. This shows that all the maps are ordered topologically correct way. That is similar genomes are mapped close to each other than the dissimilar genomes.

5 Discussion on Results

The monomer map shows some cluster formation (Fig. 3). The Feline Coronaviruses (starts with F) are clearly segregated as a cluster. The Whale Coronavirus (WH) is separated as a single element cluster.

The Avian (AV) and Turkey (TU) Coronaviruses form one cluster. There is also a separate cluster of dominating Bovine Coronaviruses (BOV, BO1, BOR) along with Human Coronaviruses 43 (H43) and 44 (H44). The Porcine (POH) and Equine (EQ) CoVs also clubbed together. Other viruses are scattered. The monomer map shows that even the coarse signature like monomer is capable of clustering some groups of Coronaviruses. This proves that these viruses do have strong identity at the very nucleotide level.

The dimer map shows a more compact cluster of the Feline Coronaviruses (Fig. 4). The only exception to this

cluster is the Alpha Corona Transmissible gastroenteritis virus (AL). In this higher order signature based clustering, the Avian and Turkey still clustered as a single unit. Whale Coronavirus is placed as neighbor to this group.

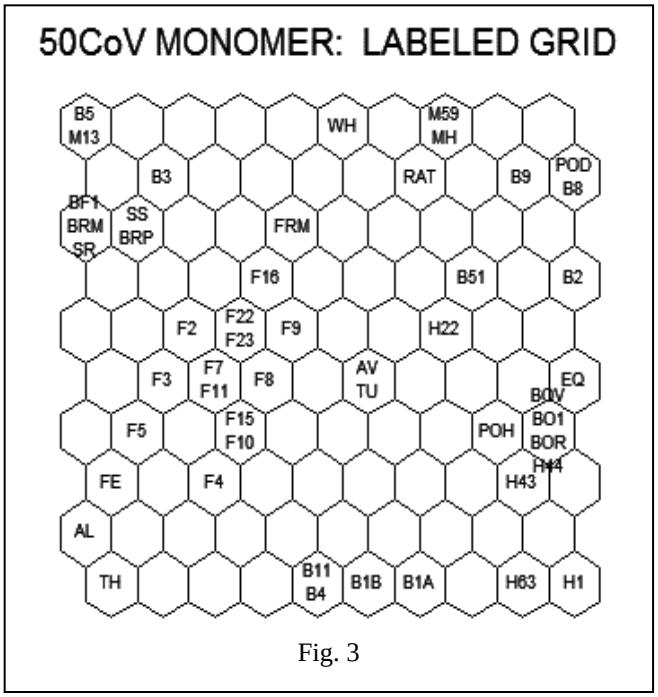


Fig. 3

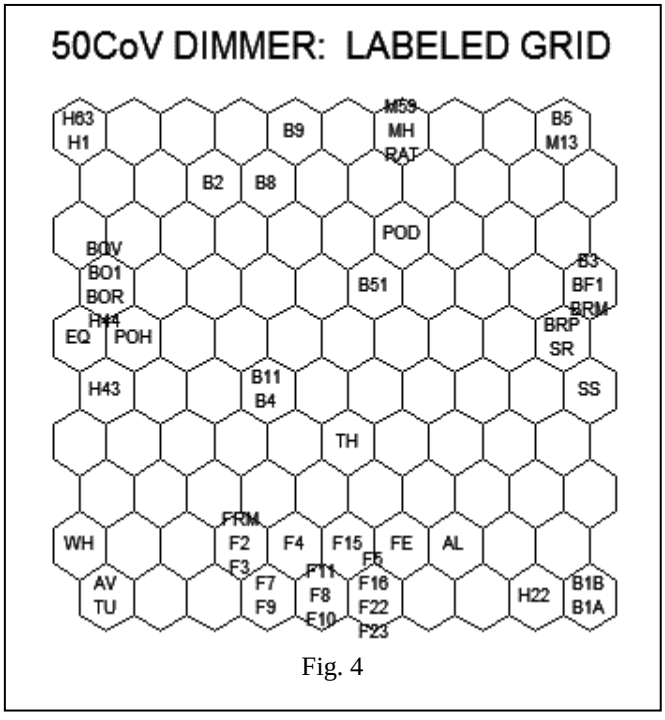


Fig. 4

The Bovine dominated cluster remains as it was in the monomer map. The SARS Coronaviruses (SR, SS) form a separate cluster along with Bat Coronaviruses (B3, BF1, BRP, BRM). The Human Coronaviruses (H1, H63, H22) are clearly separated as two clusters at the corners of the map. The bat Coronaviruses (B1A, B1B) are closely similar to the above Human Coronaviruses. This may be an indicator that the Human Coronaviruses are from these Bat Coronaviruses. The rest of Coronaviruses are scattered.

In the codon map (Fig. 5), we find three subgroups of the Feline Coronaviruses. The major Feline sub-cluster is formed with F2, F3, F4, F7, F10, F11, and FRM genomes. The second Feline subgroup F5, F9, and F15 form a separate cluster. The third Feline subgroup consists of F8, F16, F22, and F23 genomes. The connection between the Alpha Coronavirus and Feline Coronavirus continues, but only with FE genome. It may be an indicator about the possible jump of the Coronavirus between AL and Feline group of Coronaviruses.

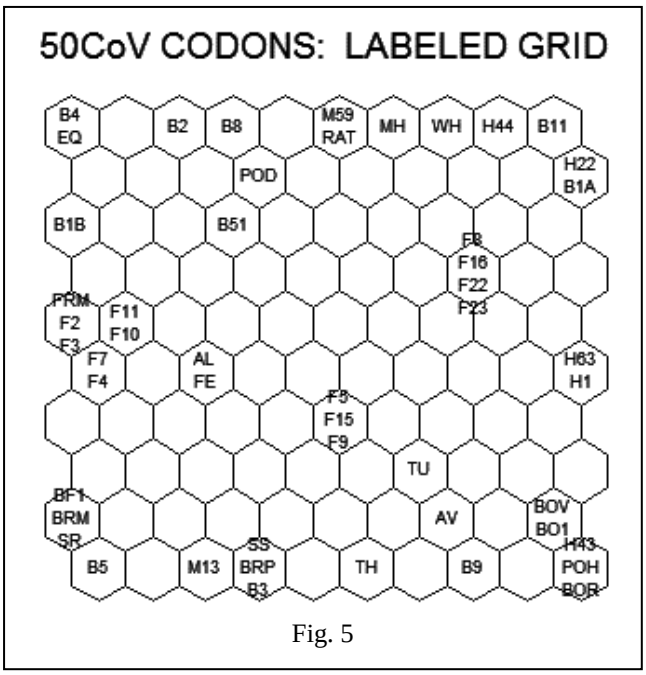


Fig. 5

The Human Coronaviruses H1 and H63 have formed a well delineated cluster having its own identity. Is this an indicator that they adapted to the human environment fully? The Human Coronavirus H22 and H44 form a separate cluster along with Bat Coronaviruses B1A and B11 genomes. The Whale, Rat, Murine Hepatitis Coronavirus MH, and the Murine Beta Coronavirus M59 are part of this cluster. The fifth Human Coronavirus (H43) is found in another cluster dominated by Bovine Coronaviruses BOV, BO1, BOR. The Porcine epidemic diarrhea virus is also found in this cluster. This may be a clue that the Porcine Coronavirus jumped to Bovine group and from there jumped to the human being as H43 genome.

The SARS Coronavirus (SS) is closely related with Bat Coronavirus B3 and the Bat SARS Coronavirus BRP. The Murine M13 is akin to this. All these 4 genomes form a single cluster. Did the human SARS Coronavirus emerge from the Bat Coronavirus BRP? Possibilities are high since the genomic signatures are very similar. The other SARS coronavirus Rs_672/2006 (SR) form a cluster of its own that

includes the Bat Coronavirus B5, BF1, and the Bat SARS CoV Rm1/2004 (BRM). The map shows clearly that there are two different subgroups of SARS Coronaviruses.

The Turkey (TU) and Avian (AV) Coronaviruses are combined with the Bat Coronavirus (B9) as a single cluster. The Porcine Coronavirus (POD) is surrounded by three different Bat Coronaviruses (B2, B8, B51) to form a separate cluster. The Equine Coronavirus (EQ) is found along with the Bat Coronavirus (B4). The Thrush Coronavirus (TH) stands as a single cluster.

The RSCU map (Fig. 6), the Human Coronavirus H1 and H63 form a separate cluster in confirmation with the Codon Map. The Human Coronavirus H22 is in a cluster along with the Bat Coronavirus B1A and B11. The Human Coronavirus H43 is integrated with Bovine Coronaviruses BOV, BO1, BOR whereas the H44 Human Coronavirus forms a separate cluster along with the Whale Coronavirus.

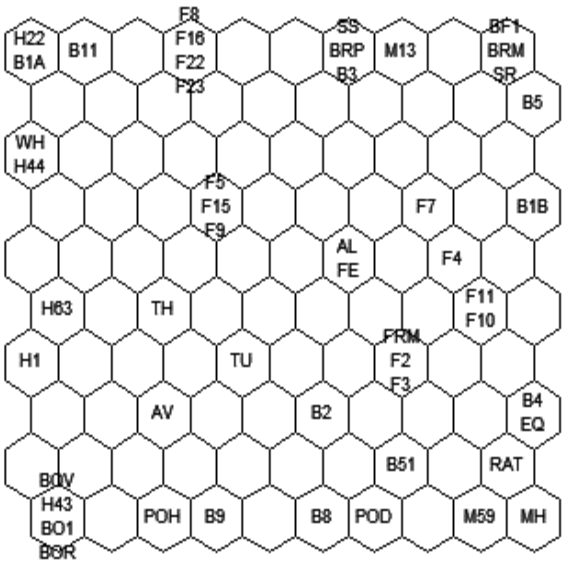


Fig. 6

The SARS Coronavirus (SS) forms a cluster along with the Bat Coronavirus B3 and BRP genomes and Munia Coronavirus (M13). The other SARS Coronavirus (SR) is clustered with the Bat Coronaviruses BF1, BRM and B5.

The Feline Coronavirus cluster F8, F16, F22, F23 and the cluster F5, F9, F15 remain the same as in the Codon map. But the third Feline group of the Codon map is divided into two clusters namely, F2, F3, FRM and F4, F7, F10, F11. The Feline Coronavirus FE remains combined with AL as it was in the Codon map.

The Thrush Coronavirus (TH) is still maintains the unique identity. But the Codon map cluster of Turkey (TU), Avian (AV), and Bat Coronavirus (B2) is separated as single unit clusters. The Porcine Diarrhea Coronavirus (POD) which was surrounded by three different Bat Coronaviruses (B2, B8, B51) to form a separate cluster in the Codon map is now combined only with B8, and B51. The Bat Coronavirus B2 is segregated as a separate single unit cluster. The Porcine Hemagglutinating Coronavirus (POH) is combined with Bat Coronavirus B9. The Murine Beta Coronavirus (M59) and the Murine Hepatitis Coronavirus (MH) combined with the Rat, Equine and Bat Coronaviruses to form a cluster.

6 Findings

From the SOM based clustering of Coronaviruses with different genomic signatures shows clearly that the Bat Coronaviruses are the cause for various other Coronaviruses. It gives a clear indication that Bat Coronaviruses are the root cause and the starting point of the spread of these viruses. The higher genomic signature based SOM maps like Codon and RSCU shows that almost all the non-Bat Coronaviruses are invariably connected with one or more Bat Coronaviruses. It points to the fact that Bat is one of highly possible reservoir of different Coronaviruses. This corroborates with the findings of Wang who reviewed studies by different groups demonstrating that SARS-CoV succeeded in spillover from a wildlife reservoir (probably bats) to human population via an intermediate host(s) and that rapid virus evolution played a key role in the adaptation of SARS-CoVs in at least two non-reservoir species within a short period [4]. Also, this fact falls in line with the findings of Shi that the horseshoe bats were identified as the natural reservoir of a group of coronaviruses that are distantly related to SARS-CoV. The genome sequences of bat SARS-like coronavirus had about 88-92% nucleotide identity with that of the SARS-CoV [19]. Graham noted that the Coronaviruses crossed to other hosts including human from zoonotic reservoirs by changes in the Spike attachment protein both within and outside of the receptor binding domain [24].

Another major finding is that the Feline Coronavirus clusters do not have any Bat Coronaviruses connected with them. This may be an indicator that this group of Coronaviruses is of different origin. This fact is supported with the following findings of Jackwood. The Sequence analysis of SARS-CoV shows that the 5' polymerase gene has a mammalian ancestry; whereas the 3' end structural genes (excluding the spike glycoprotein) have an avian origin. Spike glycoprotein, the host cell attachment viral surface protein, was shown to be a mosaic of feline coronavirus and avian coronavirus sequences resulting from a recombination event [1].

While the jumping of CoV from animals and avian to human beings is possible with recombination, it is also possible with the viral genetic mutations, according to the findings of Principi Nicola, when there is sustained exposure to the infected animals [8]. This fact is also revealed in our analysis. In the Fig.6 the WH and H44 are grouped as a well segregated cluster and the Bovine Coronaviruses BOV, BO1, BOR are clustered together neatly. Remember that these two are sumptuous human food. Further, a variation in substitution rate among lineages is more responsible than the

recombination to enable the Coronaviruses greater adaptability and to cross the species boundaries, is supported by the findings of Holmes [25].

7 Conclusion

The above experiments analyzed the 50 complete genomes of the Coronaviruses by clustering these genomes based on the different genomic signatures. The unsupervised artificial neural network SOM was used for the clustering and produced 4 different SOM maps. These maps are proved to be of invaluable tool to explore the possible roots of the different Coronavirus migrations. Further, the possible multiple reservoirs or sources of origins are indicated by the combined maps and these findings are corroborated with other findings. The experiments and the findings clearly show that SOM can be a useful tool to show the initial pointers about the path of evolution of the different types of Coronaviruses. The future work can be done on more complex genomic signatures to get a clearer picture of the origin and the evolutionary footprints of the Coronavirus types.

8 References

[1] Jackwood. "The relationship of severe acute respiratory syndrome coronavirus with avian and other coronaviruses", Avian Diseases, Vol. 50, No. 3, PP. 305-320, 2006.

[2] Marra Marco A., Jones SJM, Astell CR, Holt RA, Brooks-Wilson A, Butterfield YS, et al. "The Genome Sequence of the SARS-Associated Coronavirus", Science, Vol. 300, PP. 1399-1404, May 2003.

[3] Dijkman Ronald, M.F.Jebbink et al. "Human Coronavirus 229E Encodes a Single ORF4 Protein Between the Spike and the Envelope Genes", Journal of Virology, Vol. 3, No. 106, 2008..

[4] Wang L.R., and Eaton B.T. "Bats, Civets and the Emergence of SARS", Current Topic Microbiol. Immunology, Vol. 315, PP. 325-344, 2007.

[5] Gorbalenya AE, Snijder EJ, and Spaan WJ. "Severe Acute Respiratory Syndrome Coronavirus Phylogeny: Toward Consensus", J. Virology, Vol. 78, No. 15, pp.7863-7866, 2004.

[6] Fielding C. Burtram. "Human Coronavirus NL63: A Clinically Important Virus?", Future Microbiology, Vol. 6, No. 2, PP. 153-159, 2011.

[7] Severance G. Emily, Dickerson B. Faith, Viscidi P. Raphael, Bossis Ioannis, Stallings R. Cassie, Origoni E. Andrea, Sullens Anne, and Yolken H. Robert. "Coronavirus Immunoreactivity in Individuals With a Recent Onset of Psychotic Symptoms", Shcizophrenia Bulletin, Vol. 37, No. 1, 2011.

[8] Principi Nicola, Bosis Samantha, and Esposito Susanna. "Effects of Coronavirus Infections in Children" Emerging Infectious Diseases, Vol. 16, No. 2, Feb 2010.

[9] Kohonen Teuvo. "The Self-Organizing Maps", Springer-Verlog, 3rd Ed., Germany, 2001.

[10] Kohonen Teuvo. "Self-Organizing Formation of Topologically Correct Feature Maps", Biological Cybernatics, Vol. 43, No. 1. PP. 59-69, 1982.

[11] Kohonen Teuvo. "Self-Organizing Maps of Symbol Strings", Technical Report A 42, Helsinki University of Technology, Laboratory of Computer and Information Science, 1996.

[12] Kohonen Teuvo, and Sumervuo Panu. "Self-Organizing Maps of Symbol Strings with Application to Speech Recognition", Proceedings of WSOM, 1997.

[13] Kohonen Teuvo, Somervuo Panu. "Self-Organizing Maps of Symbol Strings", Neurocomputing, Vol. 21, PP. 19-30, 1998.

[14] Arrigo et al. "Identification of a New Motif on Nucleic Acid Sequence Data Using Kohonen's Self-Organizing Map", Bioinformatics, Vol. 7, No. 3, PP. 353-357, 1991.

[15] Kanaya, S., Kudo, Y., Abe, T., Okazaki, T., Carlos, D.C., and Ikemura, T. "Gene Classification by Self-Organization Mapping of Codon Usage in Bacteria with Completely Sequenced Genome," Genome Informatics, Vol. 9, PP. 369-371, 1998.

[16] Kanaya Shigehiko, Kinouchia Makoto, Abe Takashi, Kudoe Yoshihiro, Yamadae Yuko, Nishid Tatsuya, Morib Hirotada, Ikemuraf Toshimichi. "Analysis of Codon Usage Diversity of Bacterial Genes with a Self-Organizing Map (SOM): Characterization of Horizontally Transferred Genes with Emphasis on the E. coli O157 Genome," Gene, Vol. 276, PP. 89-99, 2001.

[17] Abe, T., Kanaya, S., Kinouchi, M., Ichiba, Y., Kozuki, T. and Ikemura, T. "A Novel Bioinformatic Strategy for Unveiling Hidden Genome Signatures of Eukaryotes: Self-Organizing Map of Oligonucleotide Frequency", Genome Informatics, Vol. 13, PP. 13-20, 2002.

[18] Abe Takashi, Sugawara Hideaki, Kanaya Shigehiko, and Ikemura Toshimichi. "A Novel Bioinformatics Tool for Phylogenetic Classification of Genomic Sequence Fragments Derived from Mixed Genomes of Uncultured Environmental Microbes", Polar Bioscience, Vol. 20, PP. 103-112, 2006.

[19] Mahony Shaun, Mclnerney O.James, Smith J. Terry, and Golden Aaron. "Gene Prediction Using Self-Organizing Map: Automatic Generation of Multiple Gene Models", BMC Bioinformatics, Vol. 5, No. 23, Mar 2004.

[20] Mahony Shaun Aengus. "Self-Organizing Neural Networks for biological Sequence Analysis", Doctoral Thesis, National University of Ireland, Galway, Dec 2005.

[21] Oja Merja. "Self-Organizing Map based Discovery and Visualization of Human Endogenous Retroviral Sequences Groups", Int. Jr. Neural Systems, Vol. 15, No. 3, PP.163-179, 2005.

[22] Kiviluoto, Kimmo. "Topology Preservation in Self-Organizing Maps, The 1996 IEEE International Conference on Neural Networks", ICNN. Part 1 (of 4); Washington DC USA; 03-06, PP. 294-299, Jun 1996.

[23] Shi Z., Hu Z. "A Review of Studies on Animal Reservoirs of the SARS Coronavirus", Virus Res. Vol. 133, PP. 74-87, No. 1, 2007.

[24] Graham L. Rachel, and Baric S. Ralph. "Recombination, Reservoirs, and the Modular Spike: Mechanisms of Coronavirus Cross-Species Tranmission", Journal of Virology, Vol. 84, No. 7, PP. 3134-3146, Apr 2010.

[25] Holmes C. Edward, and Rambaut Andrew. "Viral Evolution and the Emergence of SARS Coronavirus", Phil. Trans. Royal Society London, Biological Science, Vol. 359, PP. 1059-1065, Jun 2004.

AUTHORS BIOGRAPHY

The first author, Francis Thamburaj, is a professor in Computer Science Department of the Bharathidasan University. He has obtained the Master of Computer Application degree in 1987 and Master of Philosophy in Computer Science degree in 2001. For a decade he has been actively engaged in Artificial Neural Network research and published many papers in national and international conferences. Currently, he is an active research scholar in the inter-disciplinary area of Artificial Neural Networks and Bioinformatics.

Evolvable Lip Contour Model for Emotion Recognition

Sristi Shaw[1], Kanika Orea[1], Pavel Bhowmik[1], Anisha Halder[1], Aruna Chakraborty[2], Amit Konar[1] and Atulya K. Nagar[3]
[1] *Department of Electronics and Tele-Communication Engineering, Jadavpur University, Kolkata-32, India*
E-mail – srishti67@gmail.com,kanor30@gmail.com,bpavel88@gmail.com,
halder.anisha@gmail.com,konaramit@yahoo.co.in
[2] *Department of Computer Science and Engineering, St. Thomas' College of Engineering and Technology, Kolkata, India*
E-mail – aruna_stcet@rediffmail.com
[3] *Department. of Computer Science, Liverpool Hope University, Liverpool, UK.*
E-mail – nagara@hope.ac.uk

***Abstract*- This paper proposes an alternative approach to emotion recognition from the outer lip contour of the subjects. Subjects exhibit their emotions through their facial expressions, and the lip region is segmented from their facial images. A lip contour model has been developed, and the parameters of the model are adapted using differential evolution algorithm to match with the actual outer contour of the lip. An SVM classifier is then employed to classify the emotion of the subject from the parameter set of the subjects' lip contour. The experiment was performed on 50 subjects in the age group 20-30 years, and the worst accuracy in emotion classification is found to be 86%.**

Keywords- Image Segmentation, Support Vector Machines, Evolutionary Algorithm, Lip Contour, Emotion Recognition

I. INTRODUCTION

Emotion recognition has an important role to play in the next generation human-computer interactive systems (HCI). To aid human computer interactive systems researchers are keen to determine emotion of the subjects from their facial expression, hand gesture and voices. For instance, Ekman and Friesen [3] proposed a scheme for recognition of facial expression from the movements of cheek, chin and wrinkles. Kobayashi and Hara [7], [8], [9] developed a scheme for recognition of human facial expression using well known back propagation neural algorithm. Their scheme is capable of classifying facial expression depicting happiness, sadness, fear, anger surprise and disgust. Yamada presented a new scheme of recognizing human emotions through classification of visual information [30]. Frenandez–Dols et al. developed a scheme for decoding emotion from both facial expression and content [31]. Busso and Narayanan [11] compared the scope of facial expression, speech and multi-modal expression in emotion recognition. Cohen et al. [19], [20] considered recognition of emotions from live video using hidden Markov Model. Gao et al. [14] proposed a technique for facial expression recognition from a single facial image using line-based caricatures. Lanitis et al. [12] proposed a novel technique for automatic interpretation and coding of facial images using flexible models. Some of the other well known works of emotion recognition from facial expressions include [1], [2], [10], [13], [22], [25].

Most of the existing works on emotion recognition employ one or more of the facial attributes for recognition and interpretation about the emotional state of the subject. However, we are afraid that until this time there exists no unique facial attribute based emotion recognition scheme. This paper offers an interesting solution to this problem by considering only outer lip contour of the subject for recognition of their emotion. In [4], the authors considered a lip contour based emotion classification. They considered the lip contour as a modified ellipse. Unfortunately, lip contour has different shapes other than modified ellipse. Among other works on 2D and 3D lip modeling the works in [28]-[30] need mentioning.. This paper proposes a new method of lip contour modeling, whose parameters are better adaptable to match the model response with a given outer lip contour. Experiments reveal that static facial caricature representative of a given emotion has a unique outer lip contour [2]. Naturally, if the outer lip contour can be accurately modeled, a classifier can be employed to classify the emotion from the model parameters.

The scheme to be proposed considers a mathematical model of the outer lip contour, sufficient to capture all the necessary finer details visible on the lip boundary of the subject. An evolutionary algorithm will be employed to tune the model of the lip contour with the outer lip-boundary of a given subject. Several classifier algorithms can be utilized to study their behavioral response to map the lip parameters to emotions attempted to be synthesized on facial expression by acting. We in this paper selected a support vector machine classifier for its wide spread popularity and our own experience about its merit in emotion classification problem.

The rest of the paper is divided into 4 sections. Section II offers the modeling issues of the human outer lip contour. In section III, we introduce differential evolution and demonstrate its scope in determining the best tuned parameter set of the lip contour model with respect to a given subject's outer lip boundary, carrying a specific emotion. Section IV presents a scheme for classification of emotion of a subject from the exact parameters of her lip contour. The conclusions are listed in section V.

II. MODELING OF THE GENERALIZED LIP CONTOUR

Until this time, there is no universally accepted model of lip contours. In this paper, we start with the elementary kiss curve (Fig. 1), and modify it at different segments, as indicated in Fig. 2 to obtain an ideal model of the curve, capable of capturing most of the Indian lip contours in different emotional states.

The basic equation of kiss curve (Fig. 1) is given by

$$y^2 = (1-x^2)^3,\ -1 \le x \le 1 \qquad (1)$$

The curve returns both positive/negative values of y for each value of x. We, however, use the entire positive half of the curve, and a portion of the negative half. The remaining negative half is replaced by a parabola for better matching with lip

contours of the subjects. When the domain $-1 \le x \le 1$ is replaced by $[-l, l]$, expression (1) is written as

$$y = \left(1 - \left(\frac{x}{l}\right)^2\right)^{3/2} \qquad (2)$$

To determine all except the GA segment in Fig. 2, we scaled the right hand side of (2) and added one or more extra terms, as needed, and determine the parameters of the new curve by setting suitable boundary conditions as listed in Table-I. The resulting parameters are also given in the table.

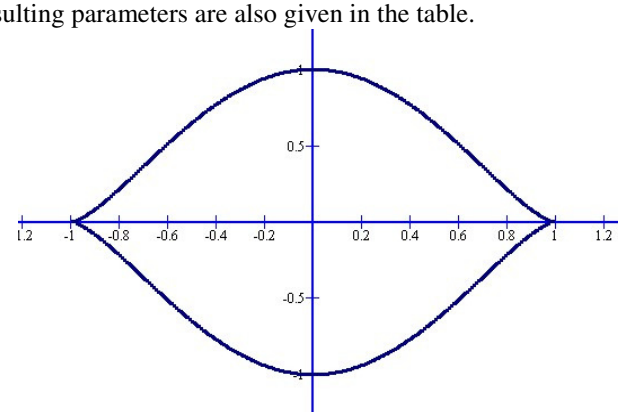


Figure 1. The standard kiss curve.

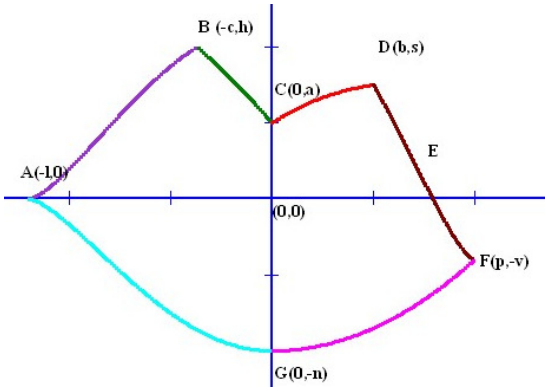


Figure 2. The proposed model of the lip outer profile.

III. SEGMENTATION OF LIP CONTOUR

Several segmentation algorithms for lip segmentation are available in the literature [1], [5]. In this paper, we, however, employ fuzzy c-means clustering algorithm to segment the lip region from the rest of the facial expression. Any pixel x_k here is considered to fall either in the Lip or the non-lip region.

Let $L(x_k)$ and $NL(x_k)$ be the membership of pixel x_k to fall in the lip and the non-lip regions respectively.

A pixel in this paper is denoted by five attributes: three attributes of color information (L*a*b) and two attributes of position information (x, y). The objective of the clustering algorithm is to classify the set of 5-dimensional data points into two classes/partitions --- the lip region and the non-lip region. Initial membership values are assigned to each 5-dimensional pixel, such that the sum of the memberships in the two regions is equal to one. That is, for the k^{th} pixel x_k ,

$$L(x_k) + NL(x_k) = 1 \qquad (3)$$

Given the initial membership values of L (x_k) and NL(x_k) for k = 1 to n^2 (assuming that the image is of size n × n), we use the FCM algorithm to determine the cluster centers, V_L and V_{NL}, of the lip and the non-lip regions:

$$V_L = \sum_{k-1}^{n^2} [L(x_k)]^m x_k \Big/ \sum_{k=1}^{n^2} [L(x_k)]^m \qquad (4)$$

$$V_{NL} = \sum_{k-1}^{n^2} [NL(x_k)]^m x_k \Big/ \sum_{k=1}^{n^2} [NL(x_k)]^m \qquad (5)$$

Expressions (2) and (3) provide centroidal measures of the lip and non-lip clusters, evaluated over all data points x_k for k=1 to n^2. The parameter m (>1) is any real number that affects the membership grade. The membership values of pixel x_k in the image for the lip and the non-lip regions are obtained from the following formulae:

$$L(x_k) = \left(\sum_{j=1}^{2} \left\{ \frac{\| x_k - v_L \|^2}{\| x_k - v_J \|^2} \right\}^{1/(M-1)} \right)^{-1} \qquad (6)$$

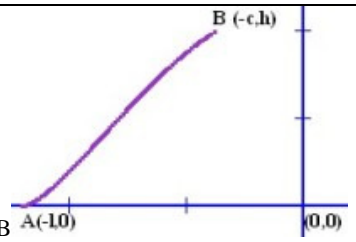
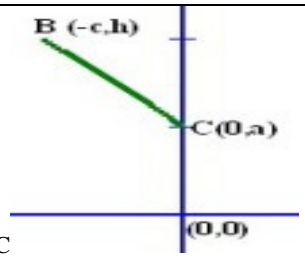
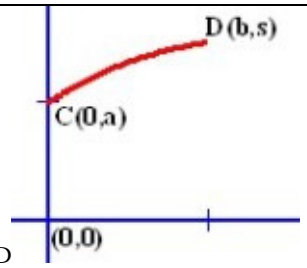
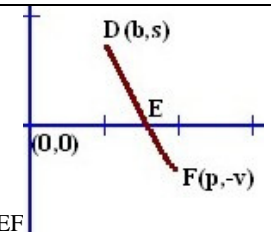
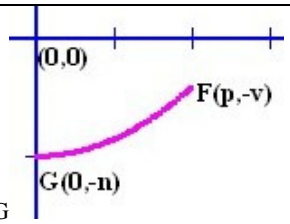
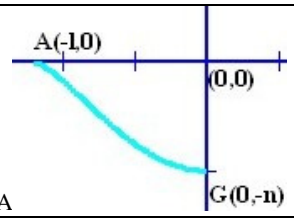
$$NL(x_k) = \left(\sum_{j=1}^{2} \left\{ \frac{\| x_k - v_{NL} \|^2}{\| x_k - v_J \|^2} \right\}^{1/(M-1)} \right)^{-1} \qquad (7)$$

where v_j denotes the j-th cluster center for j∈ {L, NL}.

Determination of the cluster centers (by (2) and (3)) and membership evaluation (by (4) and (5)) are repeated several times following the FCM algorithm until the positions of the cluster centers do not change further.

Fig. 3(a) presents a section of a facial image with a large mouth opening. This image is passed through a median filter and the resulting image is shown in Fig. 3(b). Application of the FCM algorithm to the image in Fig. 3(b) yields the image in Fig. 3(c). The dark part in Fig. 3(c) represents the lip region, and the skin and the teeth regions are represented by white color.

TABLE I. PARAMETRIC EQUATIONS FOR THE PROPOSED LIP SEGMENTS

Part of the proposed Curve	Presumed Equation	Boundary Condition	Parameters Obtained by Setting Boundary ions
 AB	$y = a_1\left(1-\left(\frac{x}{l}\right)^2\right)^{\frac{3}{2}} + a_2$	$-l \le x \le -c$ $0 \le y \le h$	$a_1 = \frac{h}{\left(1-\left(\frac{c}{l}\right)^2\right)^{\frac{3}{2}}}$ $a_2 = 0$
 BC	$y = a_3\left(1-\left(\frac{x}{l}\right)^2\right)^{\frac{3}{2}} + a_4 x$	$-c \le x \le 0$ $h \le y \le a$	$a_3 = a$ $a_4 = \frac{\left(a\left(1-\left(\frac{c}{l}\right)^2\right)^{\frac{3}{2}} - h\right)}{c}$
 CD	$y = a_5\left(1-\left(\frac{x}{p}\right)^2\right)^{\frac{3}{2}} + a_6 x$	$0 \le x \le b$ $a \le y \le s$	$a_5 = a$ $a_6 = \frac{\left(s - a\left(1-\left(\frac{b}{p}\right)^2\right)^{\frac{3}{2}}\right)}{b}$
 DEF	$y = a_7\left(1-\left(\frac{x}{p}\right)^2\right)^{\frac{3}{2}} + a_8$	$b \le x \le p$ $s \le y \le -v$	$a_7 = \frac{s+v}{\left(1-\left(\frac{b}{p}\right)^2\right)}$ $a_7 =$ $a_8 = -v$
 FG	$y = a_9 x^2 + a_{10} x + a_{11}$	$p \le x \le 0$ $-v \le y \le -n$	$a_9 = (n-v)/p^2$ $a_{10} = 0$ $a_{11} = -n$
 GA	$y = \pm\left(1-x^2\right)^{3/2}$	$0 \le x \le -l$ $-n \le y \le 0$	$y = -n\left(1-\left(x/l\right)^2\right)^{3/2}$

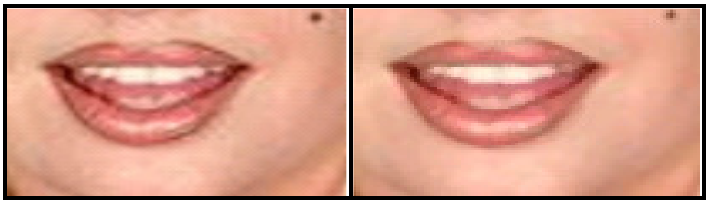


Figure 3. (a). The original face. Figure. 3. (b). The median filtered (a).

Figure. 3.(c). The segmented mouth region obtained from (b) by FCM algorithm.

IV. Parameter Extraction of a Given Lip Contour Using Differential Evolution Algorithm

Evolution Algorithms are popularly being used over the last few decades for handling unconstrained/constrained optimization problem. Given the lip contour of a subject, determining the parameters of the mathematical model representing the generalized lip contour that matches bets with the lip contour is the problem of concern in the present context. We consider a marked point of interest in a given lip outer stylus as indicated in Fig. 2. We construct a vector of 9 dimensions, representing a trial solution of the present problem. Differential Evolution (DE) proposed by Storn and Price [16] is one derivative free optimization algorithm, which offers promising solution to a global optimization problem. In this paper we employ differential evolution as the optimization algorithm to determine the lip parameters of a given subject carrying a definite emotion,

A. The classical Differential Evolution Algorithm

The DE algorithm initializes a set of trial solutions, called parameter vector. The parameter vectors are evolved through a process of mutation and recombination, and a selection scheme is used to identify the better candidate between the evolved and the trial solution (parameter vector). A brief overview of the DE algorithm is available in any standard text. We here provide a simple pseudo code to the classical DE algorithm. In this problem we used a variant of DE named as DE/rand/1/bin.

An iteration of the classical DE algorithm consists of the four basic steps – initialization of a population of vectors, mutation, crossover or recombination and finally selection. The main steps of classical DE are given below.

I. Set the generation number t=0 and randomly initialize a population of NP individuals $\vec{P}_t = \{\vec{X}_1(t), \vec{X}_2(t), ..., \vec{X}_{NP}(t)\}$ with $\vec{X}_1(t) = \{x_{i,1}(t), x_{i,2}(t), ..., x_{i,D}(t)\}$ and each individual uniformly distributed in the range $[\vec{X}_{\min}, \vec{X}_{\max}]$, where $\vec{X}_{\min} = \{x_{\min,1}, x_{\min,2}, ..., x_{\min,D}\}$ and $\vec{X}_{\max} = \{x_{\max,1}, x_{\max,2}, ..., x_{\max,D}\}$ with $i = [1,2,...., NP]$.

II. **while** stopping criterion is not reached, **do**

for $i = 1 to\ NP$

II. a **Mutation:**

Generate a donor vector $\vec{V}(t) = \{v_{i,1}(t), v_{i,2}(t), ..., v_{i,D}(t)\}$ corresponding to the i^{th} target vector $\vec{X}_1(t)$ by the following scheme $\vec{V}_1(t) = \vec{X}_{r1}(t) + F * (\vec{X}_{r2}(t) - \vec{X}_{r3}(t))$

Where r_1, r_2 and r_3 are mutually exclusive random integers in the range [1, NP]

II. b **Crossover:**

Generate trial vector $\vec{U}_i(t) = \{u_{i,1}(t), u_{i,2}(t), ..., u_{i,D}(t)\}$ for the i^{th} target vector $\vec{X}_1(t)$ by binomial crossover as

$$\vec{u}_{i,j}(t) = \vec{v}_{i,j}(t) \ if\ rand(0,1) < Cr$$
$$= \vec{x}_{i,j}(t) \quad otherwise$$

II. c **Selection:**

Evaluate the trial vector $\vec{U}_i(t)$

if $f(\vec{U}_i(t)) \le f(\vec{X}_i(t))$,

then $\vec{X}_i(t+1) = \vec{U}_i(t)$

$$f(\vec{X}_i(t+1)) = f(\vec{U}_i(t))$$

end if

end for

II. d Increase the counter value $t = t + 1$.

end while

The parameters used in the algorithm namely scaling factor '*F*' and crossover rate '*Cr*' should be initialized before calling the 'while' loop. The terminate condition can be defined in many ways, a few of which include

i) fixing the number of iterations *N*.
ii) when best fitness of population does not change appreciably over successive iterations.
iii) Either of (i) and (ii), whichever occurs earlier

B. Experiments with Differential Evolution for Parameter Tuning of Lip contour

The experiment was performed with 50 subjects where each subject aroused five basic emotions: anger, happiness, sad, relax, disgust and the movies representing their emotional experience was captured, and analyzed. Ten snapshots of each subject for each emotion were taken. The performance of differential evolution in parameter tuning of lip contour model is given table III and table V for two subjects, the corresponding facial expressions are given in Table II and IV respectively.

V. EMOTION CLASSIFICATION FROM MEASURED PARAMETERS OF LIP CONTOUR MODEL

It is noted from a large number of lip contour instances that there exist at least two parameters of the lip model clearly distinctive of individual emotions. So any typical machine learning/statistical classifier can be employed to classify the different emotional status from the parameter of lip contour. In this paper we use Support Vector Machine (SVM) [2] classifier for emotion classification from the lip data.

A Support Vector Machine (SVM) has been successfully used for both linear and non-linear classification. However, as non-linear operation yields results with lesser accuracy, in this paper, we focus on the linear operation only. To understand the basic operation of SVM, let us consider Fig. 4 where X is the input vector and y is the desired scalar output that can take +1 or -1 values, indicating linear separation of the pattern vector X.

The function f (X, W, b) can be represented as follows:

$$f(X, W, b) = \text{sign}(WX + b) \quad (8)$$

where $W=[w_1 \ w_2 \ \dots \ w_n]$ is the weight vector,

$X=[x_1 \ x_2 ... x_n]^T$ represents the input vector, b is the bias.

The function f classifies the input vector X into two classes denoted by +1 or -1. The straight line that segregates the two pattern classes is usually called a hyperplane. Further, the data points that are situated at the margins of the two boundaries of the linear classifier are called support vectors. Fig. 3 describes a support vector for a linear SVM

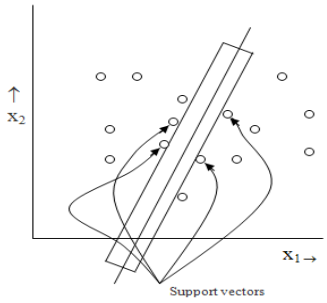


Figure 4. Defining support vector for a linear SVM system.

Let us now select two points X+ and X- as two support vectors. Thus by definition

$$WX^+ + b = +1 \quad (9.a)$$

and $$WX^- + b = -1 \quad (9.b)$$

which jointly yields

$$W(X^+ - X^-) = 2 \quad (9.c)$$

Now, the separation between the two support vectors lying in the class +1 and class -1, called marginal width is given by

$$M = \{(WX^+ + b) - (WX^- + b)\}/\|W\| = 2/\|W\| \quad (10)$$

The main objective in a linear Support Vector Machine is to maximize M, i.e., to minimize ||W||, which is same as minimizing ½WTW. Thus, the linear SVM can be mathematically described by:

Minimize $\varnothing(W) = \frac{1}{2} W^T W$ *subject to* $y_i (WX_i + b) \geq 1$ for all i, where y_i is either 1 or -1 depending on the class which X_i belongs to. (11)

Here, the objective is to solve W and b that satisfies the above equation. In this paper, we are not presenting the solution to the optimization problem, referred to above. This is available in standard text in neural network [11]. One important aspect of SVM is the kernel function selection. For linear SVM, the kernel K for two data points Xi and Xj is defined by

$$K(X_i, X_j) = X_i^T X_j \quad (12)$$

In our basic scheme, we employed five SVM networks, one each for joy, anger, sadness, fear and relaxation. The ith SVM network is trained with all of the training instances of the ith class with positive levels, and all other training instances with negative levels. The decision logic box driven by the five SVM networks ultimately recognizes the emotion corresponding to the supplied feature vector X. Fig. 5 provides this SVM-based emotion recognition scheme.

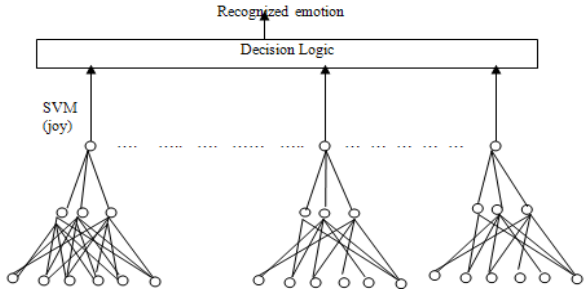


Figure 5. Emotion classification by Support Vector Machines.

To study the performance of classification using linear SVM we employed Leave-one-out cross validation. Leave-one-out classification involves using a single observation from the original sample as the classified data, and the remaining observations as the training data. This is repeated such that each observation in the sample is used once as the classified data. This is the same as a K-fold classification with K being equal to the number of observations in the original sample.

TABLE II. THREE INSTANCES OF FACIAL EXPRESSION OF SUBJECT 1

		EMOTION				
		Happy	Sad	Fear	Disgust	Anger
Instance	1					
	2					
	3					

The experiment was conducted with 50 subjects whose facial expressions representative of different emotions were captured.

From these emotional expressions, the lip region is segmented and then the lip contour identified by a DE program, and then the emotion can be classified from the extracted lip parameters using SVM.

TABLE III. LIP PARAMETERS FOR SUBJECT 1 FOR DIFFERENT EMOTIONS

		PARAMETERS								
	Instance	b	c	l	p	v	n	a	h	s
HAPPY	1	55	63	226	228	55	194	25	14	0
	2	46	80	221	249	36	241	44	52	44
	3	45	69	221	228	49	156	49	54	49
SAD	1	39	38	170	149	38	64	64	81	75
	2	37	45	171	152	38	45	45	66	53
	3	38	40	164	159	17	67	67	79	74
FEAR	1	48	49	148	142	18	116	46	60	52
	2	45	54	151	167	26	141	38	61	54
	3	44	47	143	151	29	126	36	54	44
DISGUST	1	41	40	190	168	26	96	10	24	19
	2	35	36	187	159	22	92	12	27	23
	3	39	48	176	165	30	98	12	29	18
ANGER	1	36	48	147	143	33	133	31	49	40
	2	32	48	161	168	25	134	26	45	35
	3	33	39	140	144	25	127	42	53	49

TABLE IV. THREE INSTANCES OF FACIAL EXPRESSION OF SUBJECT 2

		EMOTION				
		Happy	Sad	Fear	Disgust	Anger
Instance	1					
	2					
	3					

TABLE V. LIP PARAMETERS FOR SUBJECT 2 FOR DIFFERENT EMOTIONS

		PARAMETERS								
	Instance	b	c	l	p	v	n	a	h	s
HAPPY	1	28	36	225	197	33	139	66	76	73
	2	36	46	231	223	27	179	14	25	16
	3	37	36	221	211	0	155	58	52	65
SAD	1	38	45	149	148	17	90	61	68	64
	2	35	47	145	149	14	78	59	68	65
	3	35	40	158	146	23	74	61	75	70
FEAR	1	40	50	145	145	14	115	68	73	110
	2	28	45	136	152	19	80	23	35	28
	3	40	187	138	139	10	120	73	63	74
DISGUST	1	38	54	171	145	30	82	30	41	33
	2	37	73	173	132	37	84	24	35	28
	3	44	65	179	159	36	86	18	28	21
ANGER	1	48	40	186	185	19	149	71	81	81
	2	43	34	169	154	8	150	76	83	84
	3	45	42	158	172	19	148	54	65	63

From the expressive faces, shown in TABLE II and TABLE IV we segment only the lip region to get the lip parameters of these subjects for the five emotions. The Lip Parameters for the facial images of TABLE II and IV are given in TABLE III and TABLE V respectively.

After classification of emotion from the lip contour, it must be checked whether the emotions classified are correct or not. TABLE VI, TABLE VIII and TABLE X give a comparative study over original emotions and classified emotions. We can see in TABLE VI, for Subject-1, emotion 'Anger' is misclassified as 'Disgust' and 'Fear' at a misclassification rate of 10% and 30%, so, the correct classification rate for Anger is around 60%. Likewise, the correct classification rates of emotion Disgust, Fear, Happy and Sad for Subject 1 are 90%, 90%, 100% and 90% as listed in TABLE VI. So, the average accuracy of SVM classification for subject-1 is 86%. Thus TABLE VIII and TABLE X list the classification and misclassification rates of subject 2 and subject 3 respectively and the overall average accuracy of SVM classification is found to be 94% and 94%.

Now, since we are trying to determine the correct emotion from the lip contour only, we have to be very cautious for choosing the lip parameters. Now, we are going examine which are the most significant parameters between b, c, l, p, v, n, a, h and s for emotion classification. This is undertaken by dropping one parameter at a time, and then determining the percentage accuracy in classification. TABLES VII, IX and XI provide the results in classification accuracy obtained by dropping one parameter at a time. It is clear from the Tables that p and v are less important compared to other parameters, as in absence of them the classification accuracy does not degrade significantly, rather the average accuracy increases.

TABLE VI. COMPARATIVE STUDY OF ORIGINAL EMOTION AND CLASSIFIED EMOTION FOR SUBJECT 1

		Original Emotion				
		Anger	Disgust	Fear	Happy	Sad
Classified Emotion	Anger	60	10	10	0	0
	Disgust	10	90	0	0	10
	Fear	30	0	90	0	0
	Happy	0	0	0	100	0
	Sad	0	0	0	0	90

TABLE VII. ACCURACY OF EMOTION CLASSIFICATION AFTER OMITTING PARAMETERS ONE-BY-ONE FOR SUBJECT 1

Parameter	b	c	l	p	v	n	a	h	s
Accuracy	78	88	88	90	90	70	90	86	86

TABLE VIII. COMPARATIVE STUDY OF ORIGINAL EMOTION AND CLASSIFIED EMOTION FOR SUBJECT 2

		Original Emotion				
		Anger	Disgust	Fear	Happy	Sad
Classified Emotion	Anger	100	0	0	0	0
	Disgust	0	100	20	0	10
	Fear	0	0	80	0	0
	Happy	0	0	0	100	0
	Sad	0	0	0	0	90

TABLE IX. ACCURACY OF EMOTION CLASSIFICATION AFTER OMITTING PARAMETERS ONE-BY-ONE FOR SUBJECT 2

Parameter	b	c	l	p	v	n	a	h	s
Accuracy	86	94	96	94	90	96	90	92	88

TABLE X. COMPARATIVE STUDY OF ORIGINAL EMOTION AND CLASSIFIED EMOTION FOR SUBJECT 3

		Original Emotion				
		Anger	Disgust	Fear	Happy	Sad
Classified Emotion	Anger	100	0	0	0	0
	Disgust	0	80	0	0	0
	Fear	0	20	90	0	0
	Happy	0	0	10	100	0
	Sad	0	0	0	0	100

TABLE XI. ACCURACY OF EMOTION CLASSIFICATION AFTER OMITTING PARAMETERS ONE-BY-ONE FOR SUBJECT 3

Parameter	b	c	l	p	v	n	a	h	s
Accuracy	96	94	96	98	98	88	88	92	94

VI. CONCLUSION

The paper proposed a new approach to emotion classification from the lip contour of the subjects experiencing a specific emotion. The lip model used here is unique and unknown to the world of machine intelligence community. Experiments with 50 subjects confirm that the proposed model can capture most of the lip contour for a specific emotive experience of the subject. The DE algorithm used here is very fast and robust and thus can easily determine the parameters of the lip contour within first 50 iterations of the execution of the DE program. The SVM classifier, which is already an established tool for pattern classification with high accuracy, has been utilized here for classifying lip parameters onto emotions. Experiments here too confirm that the percentage accuracy in classification of emotion on an average is 92.4% as obtained from the data set of 50 Indian subjects each having ten frames per emotion.

REFERENCE

[1] A. Chakraborty, A. Konar, U. K. Chakraborty, and A. Chatterjee "Emotion Recognition From Facial Expressions and Its Control Using Fuzzy Logic" in *IEEE Transactions On Systems, Man, And Cybernetics Part A*, July, 2009

[2] A. Konar and A. Chakraborty, Emotional Intelligence: A Cybernetic Approach, Springer, Heidelberg, 2009

[3] P. Ekman and W. V. Friesen: Unmasking the Face: A Guide to Recognizing Emotions From Facial Clues, NJ: Prentice-Hall, 1975.

[4] M. Rizon, M. Karthigayan, S. Yaacob and R. Nagarajan, "Japanese face Emotions Classification Using Lip Features in Geometric Modelling and Imaging (GMAI'07), IEEE Computer Society, 2007.

[5] C. Bouvier, P.-Y. Coulon, X. Maldague, "Unsupervised Lips Segmentation Based on ROI Optimisation and Parametric Model", GIPSA_lab, INPG, CNRS, UJF, U.Stendhal 46 av. F. Viallet, France and LVSN, University Laval, Sainte-Foy,mQuebec, Canada, in IEEE 2007.

[6] L. Y. D. Minghui, "Lip Contour Extraction Based on Manifold" in International Conference on MultiMedia and Information Technology IEEE 2008.

[7] H. Kobayashi and F. Hara: "The recognition of basic facial expressions by neural network," Trans. Soc. Instrum. Contr. Eng., vol. 29, no. 1, 1993.

[8] H. Kobayashi and F. Hara: "Measurement of the strength of six basic facial expressions by neural network," Trans. Jpn. Soc. Mech. Engg. (C), vol. 59, no. 567, pp. 177–183, 1993.

[9] H. Kobayashi and F. Hara: "Recognition of mixed facial expressions by neural network," Trans. Jpn. Soc. Mech. Engg. (C), vol. 59, 1993.

[10] S. Bashyal, G. K. Venayagamoorthy : " Recognition of facial Expressions Using Gabor Wavelets and learning vector quantization," in International Journal of Engg. applications of Artificial Intelligence, Vol.21, 2008.

[11] C. Busso and S. Narayanan : "Interaction between speech and facial gestures in emotional utterances: A single subject study," IEEE Trans Audio Speech Language Process., vol. 15, no. 8,, Nov. 2007.

[12] A. Lanitis, C. J. Taylor, and T. F. Cootes : "Automatic interpretation and coding of face images using flexible models," IEEE Trans. PAMI, vol. 19, no. 7, pp. 743–756, Jul. 1997.

[13] Z. Zeng, Y. Fu, G. I. Roisman, Z. Wen, Y. Hu, and T. S. Huang, "Spontaneous emotional facial expression detection," Int. Jr. Multimedia, vol. 1, no. 5, pp. 1–8, Aug. 2006.

[14] Y. Gao, M. K. H. Leung, S. C. Hui, and M.W. Tananda: "Facial expression recognition from line-based caricatures," IEEE Trans. System Man Cybern. Part-A, Syst., Humans, vol. 33, no. 3, May 2003.

[15] M. Panti and I. Patras, "Dynamics of facial expression: recognition of facial actions and their temporal segments from face profile image sequences", in IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics, April 2006.

[16] K. V. Price, R. M. Storn, J. A. Lampinen, Differential Evolution: A Practical Approach To Global Optimization, Springer, 2005.

[17] L. Wang (Ed.) Support Vector Machines: Theory and Applications (Studies in Fuzziness and Soft Computing), Springer, Nov. 2010.

[18] Y. Yang, X. Wang, ,Y. Qiarr, S. Lin,"Accurate and Real-time Lip Contour Extraction Based on Constrained Contour Growing", in IEEE 2009.

[19] I. Cohen: "Facial expression recognition from video sequences," M.S. thesis, Univ. Illinois Urbana-Champaign, Dept. Elect. Eng, IL, 2000.

[20] I. Cohen, N. Sebe, A. Garg, L. S. Chen, and T. S. Huang: "Facial expression recognition from video sequences: Temporal and static modeling," Computer Vis. Image Understanding., vol. 91, Jul. 2003.

[21] M. Karthigayan, M. Rizon, Sazali Yaacob, R. Nagarajan, M. Sugisaka M. Rozailan Mamat, and Hazry Desai, "Fuzzy Clustering for Genetic Algorithm Based Optimized Ellipse Data in Classifying Face Emotion" , in International Conference on Control , Automation and Systems 2007.

[22] D. Goleman, Emotional Intelligence, New York: Bantam, 1995.

[23] S. Das, A. Halder, P. Bhowmik, A. Chakraborty, A. Konar, R. Janarthanan," A Support Vector Machine Classifier of Emotion from Voice And Facial Expression Data" in IEEE sponsored NABIC, 2009.

[24] S. Das, A. Halder, P. Bhowmik, A. Chakraborty, A. Konar, A. K. Nagar," Voice and Facial Expression Based Classification of Emotion Using Linear Support Vector Machine", in IEEE sponsored DESE, 2009.

[25] M. Ghosh, A. Chakraborty, A. Acharya, A. Konar, B. K. Panigrahi, "A Recurrent Neural Model for Parameter Estimation of Mixed Emotions from Facial Expressions of the Subjects," Int. Conf. on Neural Net, 2009

[26] M. Hisagi, T. Saitoh and R. Konishi, "Analysis of Efficient Feature for Japanese Vowel Recognition", ,in International Symposium on Intelligent Signal Processing and Communications (ISPACS '06), 2006.

[27] R.G¨ocke and A. Asthana, "A Comparative Study of 2D and 3D Lip Tracking Methods for AV ASR", Auditory-Visual Speech processing (AVSP). Moreton Island, Australia, pp.235-240, 2008.

[28] L. Revéret, and C. Benoît, "A new 3D lip model for analysis and synthesis of lip motion in speech production". in Auditory-visual Speech Processing Workshop. Terrigal, Australia. pp. 207-212,1998.

[29] T. Kuratate, K.Hsu, M.Riley, "Creating Speaker Specific 3D Lip Models Using 3D Range Data", Information Proc. Soc. of Japan, no.16, 2004.

[30] H.Yamada, , "Visual information for categorizing facial expression of emotion," Applied Cognitive Psychology, vol. 7, pp. 252-270, 1993.

[31] J. M. Fernandez-Dols, H.Wallbott,F. Sanchez, "Emotion category Accessibility and the Decoding of Emotion from Facial Expression and Context," Journal of Nonverbal Behavior, Vol.15, 1991.

SESSION

LEARNING METHODS AND RELATED ISSUES + MACHINE LEARNING

Chair(s)

TBA

Applying Context-Based Prediction in Adversarial Watkins' Q(λ)-Learning

Arisoa S. Randrianasolo and Larry D. Pyeatt
Department of Computer Science, Texas Tech University, Lubbock, TX 79409

Abstract— *This paper exhibits the transformation of Watkins' $Q(\lambda)$ learning algorithm into an adversarial Q-learning algorithm. A method called context-based prediction, borrowed from multimedia data coding, is used as opponent modeling and is incorporated in the transformed, $CBQ(\lambda)$ algorithm. We tested $CBQ(\lambda)$ by playing against three opponents. The first opponent had no prior knowledge and discovered policies as it played. The second opponent carried prior knowledge and used a fixed policy. The third opponent carried prior knowledge and continued to improve its prior policy as play progressed. $CBQ(\lambda)$ performed well by playing tic-tac-toe against the three opponents listed previously, when given 10 seconds of simulation time.*

Keywords: Reinforcement Learning, Opponent Modeling, General Game Playing.

1. Introduction

Transforming, Reinforcement Learning (RL) algorithms, that are generally used in a single player environment, into adversarial algorithms is a common interest among the Artificial Intelligence researchers [1]. In this paper, the transformation will be focused on one of the RL algorithms, Watkins' $Q(\lambda)$ learning algorithm.

Watkins' $Q(\lambda)$ has been designed to handle single agent environment. Details of the expansions and modifications to transform this algorithm to handle adversarial environment will be given in this paper. The expanded approach will be put in a situation where it can only perform simulation between moves under a short time limit.

The transformed Watkins' $Q(\lambda)$ is not guaranteed to perform well under a very short simulation time. Considerable amount of simulations are needed for the algorithm to converge into an optimal or near optimal policy. This is where adding an opponent modeling approach becomes helpful. Opponent modeling helps to improve the policy provided by the short simulation.

The adversarial Watkins' $Q(\lambda)$ learning is expanded to incorporate an opponent modeling approach in its simulation. A procedure used to perform data prediction in the multimedia domain, called Context-Based Prediction, is employed in this additional expansion as an opponent modeling approach. The resulting algorithm is tested on the trivially solvable tic-tac-toe game against other variations of the Watkins' $Q(\lambda)$ learning algorithm.

2. Watkins' Q Learning

Watkins' $Q(\lambda)$ is a variation of the Q-learning method. In Q-learning, the learning or gaming environment can be represented by a tree-like graph. The nodes of the tree represent the states in the environment. Each branch from each node of the tree describes an action that can be taken from the state represented by the node. The tree has starting states and final or goal states. There is a reward, r, associated with taking an action, a, from a state, s. Q-values are created to describe the quality of taking an action, a, while in a state, s. This value is represented by $Q(s, a)$. If the environment is in a state, s, and the learner took an action, a, then the environment transitions into a new state, s', where the learner can take a new action, a'.

The goal of the learning is to visit all the possible pairs of state-action, in the tree, multiple times in order to build a policy. A policy is a mapping from states to actions that will maximize or minimize the long term reward of the learner. In Q-learning, the learner has to finish an episode, a learning instance going from a starting state into a final state, before the Q-values of the states encountered in the episode can be updated in accordance to the total reward received. This process is referred to as backup technique.

The learning phase requires the learner to visit all possible state-action pairs multiple times before an optimal policy can be produced. The Q-value of a pair of state-action, in an episode of the simulations is not known until the end of the episode. Researchers believed that the two previous facts make Q-learning time consuming and not practical on handling real time tasks. This led Watkins to create the Watkins' $Q(\lambda)$ approach. λ is called an eligibility trace parameter which allows the learner to update the Q-value of a pair state-action, in an episode, at any step within the episode rather than at the end of the episode. Watkins' $Q(\lambda)$ performs full backup once a non-greedy move is chosen in the simulation. Sutton and Barto provide more detailed explanations about this approach [2].

Algorithm 1 is the tabular approach to Watkins' $Q(\lambda)$. $Q(s, a)$ is the Q-value as defined previously. $e(s, a)$ is the eligibility trace for a state-action pair, (s, a). The eligibility trace determines how likely a pair, (s, a), can undergo learning. γ is a discount factor, and α is a learning factor.

Algorithm 1 Tabular Watkins' Q(λ)

Initialize $Q(s,a)$ arbitrarily and $e(s,a) = 0$ for all s,a
for each episode **do**
  Initialize s,a
  while s is not terminal **do**
    Take action a, observe r, s'
    Choose a' from s' using policy derived from Q
    $a^* \leftarrow \arg\max_b Q(s',b)$
    $\delta \leftarrow r + \gamma Q(s',a') - Q(s,a)$
    $e(s,a) \leftarrow e(s,a) + 1$
    for all s,a **do**
      $Q(s,a) \leftarrow Q(s,a) + \alpha\delta e(s,a)$
      If $a' = a^*$, then $e(s,a) \leftarrow \gamma\lambda e(s,a)$
      else $e(s,a) \leftarrow 0$
    end for
    $s \leftarrow s'; a \leftarrow a'$
  end while
end for

2.1 Context-Based Prediction

Context-Based prediction is a prediction technique used to encode audio and video [3]. The main idea behind this technique is to code an input symbol depending on in relationship to neighbors. The Context-Based approach can be explained by the following example. Assume the following stream of input, $xyxxyyzz$, was read by the encoder. A frequency table is created to illustrate how many times each symbol has occurred in the stream, this process creates what is called context of order-0. The frequency table is extended to also count the number of occurrences of each symbol after one another, this creates the context of order-1. The frequency can be further extended to count the occurrences of each symbol after two successive symbols, this creates the context of order-2. The order of the context can go as far the encoder wishes.

Given the frequency table, assume that the encoder has read xy and the symbol z appears next. The encoder has the choice of coding z in order-2 with the context xy or code z in order-1 with the context y or code it in the order-0 with the context z. The advantage of using context-based coding is that after reading some part of the input stream, the decoder uses the frequency table built by the encoder to predict which symbol will likely to appear next. Detlev et al. provide more detailed detailed information about this technique [3].

3. Related Work

3.1 The Equilibrium Approach

Attempts to create Adversarial Q-learning using the equilibrium approach started with the creation of Nash Q-learning [4]. Nash Q-learning is an adaptation of Q-learning for multi-agent environments. The state-action pair value is transformed from its original form, $Q(s,a)$, into $Q(s,a^1,\ldots a^n)$, to represent the actions taken by all of the players involved. Each player is assumed to own multiple policies or strategies. This approach stores a Q-value and a Nash-Q value for each player. The Nash-Q value is used in the equation that updates the Q-value. The Nash-Equilibrium point, a set of n tuple strategies or policies, is used in the equation that updates the Nash-Q value. Detailed information about this approach is presented by Hu and Wellman [4]. Other approaches based on equilibrium calculation exist. Among these approaches are Pareto Q-learning and Correlated Q-learning [5].

3.2 The UCT approach

Q-learning, in general, requires balancing exploitations and explorations so that the policy being developed converges into an optimal policy at a faster rate. To balance exploitations and explorations, researchers developed a strategy called the Upper Confidence bound applied to Tree (UCT) [6]. UCT updates the Q-value using the equation

$$Q^*_{UCT}(s,a) = Q_{UCT}(s,a) + a\sqrt{\frac{\lg n(s)}{n(s,a)}}. \quad (1)$$

It selects an action by following the calculation,

$$\pi_{UCT}(s) = \arg\max_a Q^*_{UCT}(s,a), \quad (2)$$

where $n(s,a)$ is the number of times that an action, a, has been selected from a state, s, and $n(s)$ is the total number of visits to a state, s,

$$n(s) = \Sigma_a n(s,a). \quad (3)$$

In the UCT approach, if an action, a, has not been used in a state, s, then UCT defaults to selecting this action rather than using the actions that have been used. In a game, if the resulting state, s', from taking an action, a, has not been seen during the simulation, then the UCT policy, π_{UCT}, is abandoned and replaced by a random policy.

UCT is a famous approach in solving the GO game [7] and becomes the method of choice in the General Game Playing research [8]. The winners of the AAAI General Game Playing competitions in 2007 and 2008, called CADIAPLAYER [9], and the winner of the 2009 and 2010 competitions, called ARY, were all based on the UCT approach.

3.3 Opponent Modeling in RL

Adversarial Reinforcement Learning researchers have noticed that an opponent modeling technique can improve the policy being produced. The traditional approach to opponent modeling consists of creating a frequency table that records the number of times a move was performed by an opponent for each state in the game [10]. This strategy requires numerous games to be played in order to efficiently predict the opponents. In these numerous games, the learner hopes that each state is at least visited once. The frequency table

is updated at every instance of the game. When facing a new state that has not been encountered, the frequency table either predicts the opponent's moves with a uniform distribution over the possible actions [11], or just picks a default and fixed move. The prediction power of the frequency table, therefore, is limited when facing an unvisited state.

Some opponent modeling approaches derive an opponent's tree from the Q-values of the learner [9]. In the two players version of this method, it is assumed that the learner's loss is a gain to the opponent. The derivation of the opponent's tree requires that a maximum reward be defined for each move in each state. In this case, if the learner has a Q-value of $Q(s, a)$ for a state, s, and an action, a, the opponent will have a Q-value of $maxreward(s, a) - Q(s, a)$, where $maxreward(s, a)$ is the maximum reward observed for taking action a in state s for the history of the game.

Other opponent modeling approaches assume that opponents will always choose the move that minimizes the learner's Q-value for a state, s. Numerous games are required to have been played for these approaches to be effective. The maximum reward cannot always be defined ahead of time for each state-action pair. This fact limits the scope of the opponent's tree approach. Opponents do not necessarily choose the move that minimizes the learner's Q-values in the game. They may have their own strategies as well. When facing a state that has not been visited before, these approaches cannot be used to perform prediction.

4. Adversarial Expansion

In order to use Watkins' $Q(\lambda)$ in adversarial situations, the Q-value is transformed to represent the value of the combination of multiple states and an action, $Q(s_1, \ldots, s_n, a)$. Each state, s_1 to s_n, represents the state of each agent, labeled from 1 to n, in the environment. Each agent's state reflects the moves that the agent took in the environment. Each state does not reflect the other agents or opponents' moves. However, the agents' states when combined together can reflect the general state of the environment.

In an adversarial environment, simulation time is limited. Each agent makes a move or provides a response within a limited interval of time. To accommodate this restriction, simulation is only performed during the time limit between moves. The length of the simulation depends on the imposed response time and the number of players involved.

Since response time is limited, running a simulation from the initial starting point of the game or from an empty board does not appear to be beneficial. In this adversarial expansion, the simulation starts from the game or environment's current state.

4.1 Context-based Opponent Modeling

In addition to the previous modifications, an opponent modeling technique that utilizes the context-based prediction is used in the simulation. The idea of creating an opponent model, in this approach, comes from the following observations. First, under a limited simulation time, generally accurate information about the opponents' behavior would help to generate a good policy. The policy generated is not expected to be optimal. However, it is expected to be strong enough to defeat the opponents' policy. Secondly, the opponents' moves, in most cases, follow some patterns.

4.1.1 Context update

We define a context as an ordered collection of moves performed by an opponent. Assume, for example, that at time, t, an opponent, $p1$, has performed the moves $a_1 a_2 \ldots a_k$ and just made a move a_t. It can be said that in the context of $a_1 a_2 \ldots a_k$ the opponent's move is a_t. This can be represented by the context function,C, as follows,

$$C_{p1}(a_1 a_2 \ldots a_k, a_t) = C_{p1}(a_1 a_2 \ldots a_k, a_t) + 1. \quad (4)$$

As the opponents are allowed to change behaviors, the move performed at each context can change over time. To reflect a change of the opponent's behavior, a probability table is derived from the context update function. By using the same opponent, $p1$, as above, the probability of this opponent tacking the action, a_t, while the context is $a_1 a_2 \ldots a_k$ is calculated in the following way,

$$Prob_{p1}(a_1 a_2 \ldots a_k, a_t) = \frac{C_{p1}(a_1 a_2 \ldots a_k, a_t)}{sum}, \quad (5)$$

$$sum = \sum_{i=0}^{n} C_{p1}(a_1 a_2 \ldots a_k, a_i). \quad (6)$$

4.1.2 Using Context Probability

In the simulation, the probability table, derived from the context update, is used to determine each opponent's following move. Like any ordinary frequency table, the move that has the highest probability is the most likely to occur. The following equation reflects this case. Assuming that $p1$'s current context is $a_1 a_2 \ldots a_k$,

$$a_t = \arg\max_{a_i} Prob_{p1}(a_1 a_2 \ldots a_k, a_i) \quad (7)$$

A context that has no record of the opponents' moves will be encountered in the simulation. It is in this situation that context-based prediction differs from regular frequency tables.

An opponent's context is a direct reflection of its moves. Suppose an opponent is in a context, c, where

$$c = a_1 a_2 \ldots a_k, \quad (8)$$

with a_k being the most recent move. Assume that there is no record of the actions taken by this opponent in this context, c. To predict the next move of this opponent, the least recent move is eliminated. In this case, a_1 is eliminated from c to obtain a new context,

$$c' = a_2 \ldots a_k. \quad (9)$$

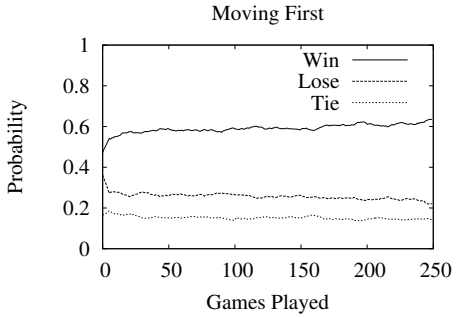
Algorithm 2 Example of Context Based Prediction

```
context c = a_1 a_2 ... a_k
opponent move a_t = null
leastrecentindex = 1
a_t = arg max_{a_i} Prob(c, a_i)
while c ≠ a_k and a_t = null do
    c = c − a_{leastrecentindex}
    leastrecentindex = leastrecentindex + 1
    a_t = arg max_{a_i} Prob(c, a_i)
end while
if a_t = null then
    a_t = randommove
end if
return a_t
```

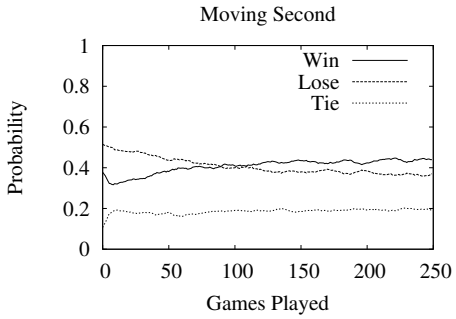
The move that is predicted to be taken by the opponent in c' can be calculated by the following formula,

$$a_t = \arg\max_{a_i} Prob_{p1}(c', a_i). \quad (10)$$

If no move has been recorded in c', then the least recent move in c', is eliminated again to obtain a new context, c''. This process keeps going if no move has been recoded at each new derived context. If certain numbers of new contexts have been derived or if the context that directly represents the last move is met and there is still no move recorded at this context, then a default action selection for the opponent is used. A default action selection for the opponent can be a random selection or some sort of ε-greedy policy. In a game of two players, for example, the opponent can default to always choose the move that will give the learner the least reward. Algorithm 2 is an example of an context-based prediction.



(a) $CBQ(\lambda)$ has first move.



(b) $CBQ(\lambda)$ has second move.

Fig. 1: Comparison of $CBQ(\lambda)$ against an RL agent that begins with no knowledge of how to play the game. The two players alternated taking the first move. The cases where the $CBQ(\lambda)$ player had the first move are shown in (a), and the cases where the other player had the first move are shown in (b).

5. Experiment

In this section, Context-Based $Q(\lambda)$, which we refer to as $CBQ(\lambda)$, is the approach described in Section 4. $CBQ(\lambda)$ was tested on the tic-tac-toe game against variations of the original Watkins' $Q(\lambda)$ algorithm. This section summarizes the performance of $CBQ(\lambda)$ against each of its opponents. $CBQ(\lambda)$ and all of its opponents used ε-greedy action selection with $\varepsilon = 0.1$. i.e. They chose the action with the highest Q-value with a 0.9 probability. The parameters γ, α, and λ were respectively set to 0.999, 0.3, and 0.9. A 9-mean smooth was used to smooth the data from the experiments.

5.1 CBQ(λ) Against a Learner

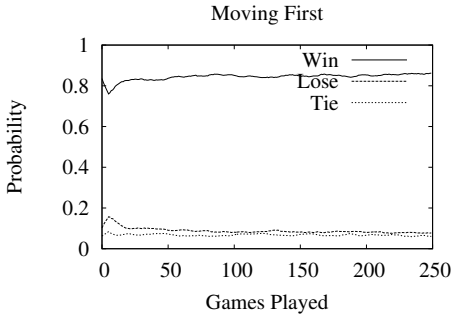
A variation of the Watkins' $Q(\lambda)$ was used as $CBQ(\lambda)$'s first opponent. This first opponent incorporated all the modifications, explained in Section 4, minus the opponent modeling technique. A total of 1000 runs were performed. Each run contained 500 games of tic-tac-toe. Both players were given 10 seconds of simulation time before making a move. The players alternated taking the first move in each game. In addition to the simulations, both players received feedback about the outcomes of their moves in the game. At the end of 500 games, the Q-values for the two players were wiped out and were reset to random values. Figure 1 summarizes the results of this experiment.

5.2 CBQ(λ) Against a Trained Player

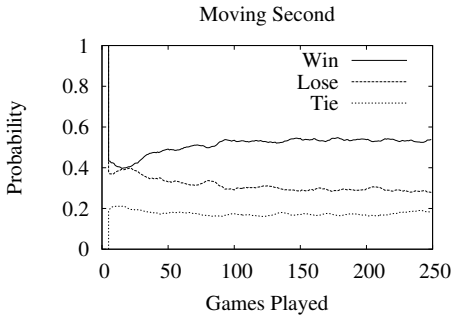
$CBQ(\lambda)$ played another 1000 runs against the original tabular version of Watkins' Q(λ). This original version was trained on 200,000 tic-tac-toe games before it played against $CBQ(\lambda)$ in each run. Recall that one run has 500 games. This original Watkins' Q(λ) did not learn from the games in the run. It used a 0.1-greedy policy that it created out of the 200,000 simulations. As in the previous experiment, players took turns to make the first move in each game. The Q-values for $CBQ(\lambda)$ and its opponent were wiped out at the end of each run and were reset to random values. Figure 2 summarizes the result of this experiment.

5.3 CBQ(λ) Against a Trained Learner

In this experiment, $CBQ(\lambda)$'s opponent was constructed from the combination of the two previous opponents. The

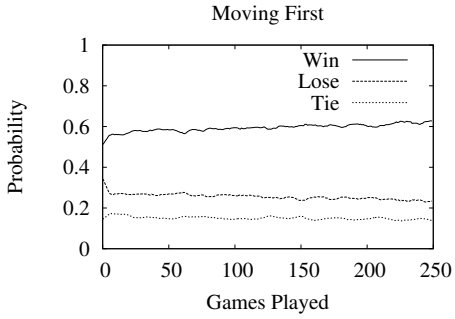


(a) $CBQ(\lambda)$ has first move.

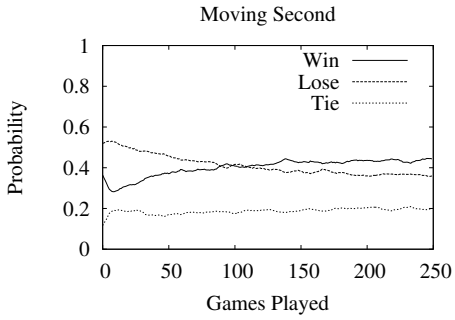


(b) $CBQ(\lambda)$ has second move.

Fig. 2: Comparison of $CBQ(\lambda)$ against an RL agent that begins with knowledge of how to play the game and ceased learning once it plays $CBQ(\lambda)$. The two players alternated taking the first move. The cases where the $CBQ(\lambda)$ player had the first move are shown in (a), and the cases where the other player had the first move are shown in (b).



(a) $CBQ(\lambda)$ has first move.



(b) $CBQ(\lambda)$ has second move.

Fig. 3: Comparison of $CBQ(\lambda)$ against an RL agent that begins with knowledge of how to play the game and continues learning while playing against $CBQ(\lambda)$. The two players alternated taking the first move. The cases where the $CBQ(\lambda)$ player had the first move are shown in (a), and the cases where the other player had the first move are shown in (b).

opponent was trained on 200000 games before each run and it also learned from each game played in the run. 1000 runs of 500 games were performed in this experiment. This experiment was conducted with the same conditions as in the previous ones. Figure 3 summarizes the outcome of this experiment.

6. Testing Results

The player that makes the first move in a tic-tac-toe game has distinct advantages over the player that makes the second move. This imposes representing each experiment in this paper by two graphs. The first graph, labeled (a), reflects the cases where $CBQ(\lambda)$ moves first. The second graph, labeled (b), reflects the cases where $CBQ(\lambda)$ moves second.

Figure 2 reflects the superiority of $CBQ(\lambda)$ against a trained player with a fixed policy. The fixed policy opponent, trained on 200,000 games, is definitely weaker compared to $CBQ(\lambda)$. Figure 2(b) shows that it is only able to challenge $CBQ(\lambda)$ during the first 20 games when $CBQ(\lambda)$ is set to make the second move. The context-based opponent modeling technique helped to capture the behavior of the static 0.1-greedy policy that the trained player was using. This situation gave $CBQ(\lambda)$ better simulations and a superior policy. $CBQ(\lambda)$ was expected to have clear advantages against fixed policy players and Figure 2 seems to confirm this claim.

The experiment in Section 5.1 was intended to measure the performance of $CBQ(\lambda)$ against a player that changes behavior. Figure 1(a) shows that $CBQ(\lambda)$'s winning probabilities are not as high as in Figure 2(a) when $CBQ(\lambda)$ is allowed to have the first move. This situation happened because the opponent is allowed to learn from the moves that happened during the games. In Figure 1(b), the crossover point where winning probabilities become greater than losing probabilities happens after playing approximately 100 games. This is due to the opponent's learning capability and the opponent's unstable policy. The context-based opponent modeling helped $CBQ(\lambda)$ to improve its winning probabilities to go above the losing probabilities despite the fact that the opponent's policy was unstable. As the opponent's policy became increasingly stable, $CBQ(\lambda)$'s winning probabilities

increased and its losing probabilities decreased. Figure 1 reflects the effectiveness of $CBQ(\lambda)$ against a player that learns at the same time as it is playing. The context-based opponent modeling captured the opponent's change of behavior and this helped $CBQ(\lambda)$ to have better simulation performance.

The experiment described in Section 5.3 consisted of confirming that the results from the two previous experiments remained correct when the two previous opponents' approaches are combined. Figure 3 shows that $CBQ(\lambda)$'s performance, in this experiment, is very similar to its performance when competing with a learning opponent (Figure 1). There is only one significant difference, and it can be seen from Figure 3(b). The exact crossover point, still appearing after hundreds of games, becomes harder to detect. For a few games, around the crossover point, $CBQ(\lambda)$'s winning and losing probabilities are equivalent. Similar to the previous experiments, $CBQ(\lambda)$'s learning capability is shown by the fact that the winning probabilities increase and become greater than the losing probabilities as more games were played. $CBQ(\lambda)$'s superiority and the context-based opponent modeling strength were strongly supported in this experiment.

7. Conclusions and Future Works

Overall, $CBQ(\lambda)$ was effective against a trained player with a fixed policy, against an untrained player which learned a policy and against a combination of both players. $CBQ(\lambda)$'s superiority against these players speculates that it can achieve high performance in an adversarial environment while only given a very short thinking time. The overall satisfying performance of $CBQ(\lambda)$ confirms that the context-based opponent modeling is appropriate to the adversarial environment described in the testing. Even with short simulation time, the context-based opponent modeling appears to make $CBQ(\lambda)$'s policy superior to its opponents.

Improvements can possibly be added to make $CBQ(\lambda)$ better. In the few games at the beginning of each run, only a few of the opponent's moves could be observed. It may not be beneficial to use context-based prediction in the early games. Executing simulations without any prediction may be beneficial in this case. A measurement function can be introduced to decide whether the context-based probability table should be used in the simulation or not. For every game, g_i, in the early part of the run, a measurement, $M(CT_{g_i})$, can be calculated. $M(CT_{g_i})$ reflects how the context-based probability table appears when game g_i is about to start. If $M(CT_{g_i})$ is bigger than some threshold, then the context-based probability table can be used starting from game g_i to the last game in the run. $M(CT_{g_i})$ shall take in consideration the contexts that have been updated and the probability values in the table. Using this measurement will allow more simulations to be done with the default action selection, while the context-based probability table has only a small amount of information.

In this paper, a single policy was used to play against the opponents. A near accurate opponent modeling shall allow the use of multiple policies to respond to the opponent's behavior. Another measurement function, M_2, can also be created to choose which policy would be used. This measurement can be done by calculating before each simulation,

$$\pi_t = \arg\max_{\pi_i} M_2(\pi_i \mid CT) \tag{11}$$

π_t will be the policy used in the simulation at time, t. Each π_i refers to each policy that the learner may have. CT is the context-based probability table. M_2 can run a full or partial episode of one game using each policy π_i and records the accumulated rewards. This approach can add more computation time and may not be practical in a situation where a very short thinking time is allowed.

References

[1] W. Uther and M. Veloso, "Adversarial reinforcement learning," in *Proceedings of the AAAI Fall Symposium on Model Directed Autonomous Systems*, 1997.

[2] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge,Massachusetts: The MIT Press, June 1998.

[3] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the h.264/avc video compression standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 620–635, 2003.

[4] J. Hu and M. P. Wellman, "Nash q-learning for general-sum stochastic games," *Journal of Machine Learning Research*, vol. 4, pp. 1039–1069, 2003.

[5] A. Greenwald, K. Hall, and M. Zinkevich, "Correlated q-learning," *Journal of Machine Learning Research*, vol. 1, 2007.

[6] S. Gelly and D. Silver, "Combining online and offline knowledge in uct," in *ICML '07: Proceedings of the 24th international conference on Machine learning*. New York, NY, USA: ACM, 2007, pp. 273–280.

[7] M. Enzenberger and M. Mueller, "An open-source framework for board games and go engine based on monte-carlo tree search," *Technical Report TR 09-08, Dept. of Computing Science. University of Alberta, Edmonton, Alberta, Canada*, Dec 2009.

[8] M. Genesereth and N. Love, "General game playing: Overview of the aaai competition," *AI Magazine*, vol. 26(2), pp. 62–72, 2005.

[9] F. Hilmar and B. Yngvi, "Simulation-based approach to general game playing," in *AAAI'08: Proceedings of the 23rd national conference on Artificial intelligence*. AAAI Press, 2008, pp. 259–264.

[10] J. Wu, C. Ye, and S. Jin, "Opponent learning for multi-agent system simulation," in *Rough Sets and Knowledge Technology*, ser. Lecture Notes in Computer Science, G. Wang, J. Peters, A. Skowron, and Y. Yao, Eds. Springer Berlin / Heidelberg, 2006, vol. 4062, pp. 643–650.

[11] H. Wang, Y. Gao, and X. Chen, "Rl-dot: A reinforcement learning npc team for playing domination games," in *Computational Intelligence and AI in Games, IEEE Transactions*, vol. 2, march 2010, pp. 17–26.

A Learning Algorithm for Question Type Classification

Richard Khoury[1]
[1]Department of Software Engineering, Lakehead University, Thunder Bay, Ontario, Canada

Abstract - *Question type (or answer type) classification is the task of determining the correct type of the answer expected to a given query. This is often done by defining or discovering syntactic patterns that represent the structure of typical queries of each type, and classify a given query according to which pattern they satisfy. In this paper, we combine the idea of using informer spans as patterns with our own part-of-speech hierarchy in order to propose both a new approach to pattern-based question type classification and a new way of discovering the informers to be used as patterns. We show experimentally that using our part-of-speech hierarchy greatly improves type classification results, and allows our system to learn valid new informers.*

Keywords: Question type classification; Natural language processing; Answer type classification; Part-of-speech; Syntactic pattern; Informer span

1 Introduction

Question type (or answer type) classification is the Natural Language Processing (NLP) task of determining the correct type of the answer expected to a given query, such as whether the answer should be a person, a place, a date, and so on. This classification task is of crucial importance in question-answering (QA) systems. Indeed, researchers from Concordia University have shown that questions that have been classified into the correct type are answered correctly twice as often as misclassified questions, since the answering system can make use of type-specific algorithms to pinpoint the answer and reject answers of the wrong type [1].

Over the years, many varied approaches to question type classification have been proposed. The simplest approaches are to detect type-specific keywords in the query [1] or to use the wh-term (who, what, where, when, why, which, whom, whose, how) at the beginning of the query for disambiguation. However, both approaches can be easily misled. Keywords alone are not enough to differentiate between question types: for example, the queries "which emperor was defeated on that date" and "on which date was that emperor defeated" have exactly the same words but clearly belong to different types. And wh-terms can take different meanings based on the context. For example, "who was Napoleon" and "who defeated Napoleon" both begin with *who* but only the second is asking for a person while the first is asking for a historical definition, and "what French emperor was defeated at Waterloo" is a *who* query phrased as a *what* query. In fact, research has shown that rephrasing a question to use a different wh-term is the single most common way that humans use to paraphrase queries [2]. In light of this, syntactic-pattern-based methods have become popular in question type classification systems [3]. These methods use or discover syntactic patterns that represent the structure of typical queries of each type, and classify a given query according to which pattern they satisfy. For example, a system could discover, by comparing example queries, that those featuring the pattern "...in which year..." should be classified in the *time* type [3].

In this paper, we combine an idea from Krishnan *et al.* of using informer spans as question type classification patterns [4] with our own part-of-speech hierarchy [5] in order to propose both a new approach to pattern-based question type classification and a new way of discovering the informers to be used as patterns. We show experimentally that using our part-of-speech hierarchy greatly improves type classification results, and moreover that it can be used to learn new valid informers from example queries.

The rest of this paper is organized as follows. In Section 2, we review a representative sample of work done on the task of question type classification in order to clearly illustrate the nature of our contribution. The theoretical frameworks of our classification system, of our learning algorithm and of our part-of-speech hierarchy are all presented in Section 3. Our ideas have all been implemented and tested, and experimental results are presented and discussed in Section 4. Finally, we offer some concluding remarks in Section 5.

2 Related Work

Question type classification is an integral task in most QA systems developed today. Consequently, there are considerable variations in the nature of the classification systems, in the set of question types recognized, and in the nature and size of the knowledge base underlying the classification systems.

A Naïve Bayes question type classifier was proposed in [6]. The authors computed the probability of a query belonging to a question type given the prior probability of that question type multiplied by the conditional probability

of the query's features (i.e. the words left after stemming and stopword removal) given that type. They used six question types for their classification: time, location, human, number, object, and description. Their system achieves an average precision of 59% over these six types. However, the authors do not discuss how large the table of conditional probabilities has to be to account for a reasonable number of features in six categories.

A team from the University of Concordia developed a simple keyword-based question type classification system as part of their QA system for the TREC-2007 competition [1]. Their system classifies queries into seven types: date, location, person, organization, number, website, and other, by recognizing keywords in the query. Their research does show that a keyword-matching approach can work quite well given a large and detailed enough lexicon: they report that their system achieves 93% accuracy. However, they do not give the size of the lexicon they used or discuss how they built it. Their comment in [1] that "the type classification was improved this year by adding about 50 new words to the word-match dictionary and modifying the algorithm" indicates that their lexicon is hundreds or thousands of words in size, that it was built over the course of years, and that the classification results is directly linked to its size and level of detail. These are all serious limitations of the keyword-matching approach.

Another important limitation of a keyword-based classifier is that, taken out of context in a bag-of-words model, keywords can easily mislead the classification. This problem was highlighted by Tomuro in his study on the impact of semantic information in question type classification [7]. Tomuro built two question type classifiers, one using a decision tree and one using a k-nearest-neighbour algorithm, and trained two instances of each classifier. The first instance used only a closed lexicon of about 100 words frequently found in queries. This lexicon is thus composed mostly of domain-independent, non-content, closed-class words and includes wh-terms. The second instance of both classifiers uses the same lexicon and also adds in the categorization (i.e. WordNet hypernym) of query words that are not part of the lexicon. The classifiers were trained to recognize 12 question types: time, location, entity, definition, reference, reason, procedure, manner, degree, atrans, interval, and yes-no questions. His results show that taking the extra keywords into account and adding the categorization information does not affect the results in a statistically significant way. Moreover, he found that the extra knowledge added can mislead the classifier when a query's syntax affects its semantic meaning. For example, the query "what does Hanukah mean?" is clearly a definition-type question, but the keyword "Hanukah" is a hyponym of "time period" in WordNet, and thus misleads his system into classifying it as a time-type question [7]. Another WordNet-based question type classifier proposed in [8] faced similar difficulties when dealing with certain queries, and overcame them by creating a two-step classification scheme. It begins by defining a set of simple question type syntactic patterns, such as "what {is|are} <phrase>?" for definition-type queries. Their system first attempts to match queries to these patterns, and only goes to WordNet hypernym searches for queries that cannot be matched. The results in [7] and [8] illustrate how syntactic patterns can be more powerful than simple keyword recognition for question type classification, as they offer a more complete picture of the query.

Zhang and Nunamaker [9] built a pattern-based question type classifier for their video indexing and retrieval system. They defined nine question types in their system: time, location, person, organization, number, object, reason, definition, and undefined. Their system classifies each user's query according to a set of simple patterns that combine both wh-term and the categories of keywords found in the query. To illustrate, a sample pattern given in [9] is "if (question starts with 'what' + person) then (answer type is person)". Krishman *et al.* take the idea of type classification patterns one step in a different direction in [4] and propose that the patterns could consist simply of a short string of contiguous words found in the query, which may or may not include wh-terms, and which they called the *informer span* (or simply *informer*) of the query. Their work shows clearly that a support vector machine classifier using hand-made informers yields better results than question bigrams, and that informers discovered automatically work almost as well as hand-made ones.

Unfortunately, the authors of [8], [9] and [4] do not give a complete list of patterns nor the total number of patterns in their respective systems. But Krishman *et al.* point out in their review that such systems are built on hundreds of unpublished patterns [4]. This large size is required because these systems look for an exact syntactic match to the user's query, and the same query can be paraphrased in a variety of very different ways [2].

3 Methodology

In this paper, we develop a new system for question type classification based on a part-of-speech hierarchy that we present below. As will become evident, using the part-of-speech hierarchy makes it possible to get good classification results using only a handful of simple informers. This stands in stark contrast to the other systems reviewed in Section 2, which need large sets of patterns, lexicons, or probability tables to work well.

The question types we use are "person" (who), "date" (when), "physical object" (what), "location" (where), "numeric value" (how many/how much), and "description" (how/why). We manually define a set of 16 simple informers to represent these question types; a trivial task given that they each have clear wh-terms. We design these

informers to follow a simple pattern: each is two words long, with one wh-term and one higher-level POS from our hierarchy that represents a past-tense verb (vbd), a present-tense verb (vbz), a description adjective (jj), a singular common noun (nn) or a plural common noun (nns). The informers are listed in Figure 1.

Who [vbz]	What [nn]	How [jj]
Who [vbd]	What [nns]	How [nn]
When [vbz]	What [vbz]	How [vbz]
When [vbd]	What [vbd]	How [vbd]
Where [vbz]	Why [vbz]	
Where [vbd]	Why [vbd]	

Figure 1: 16 basic informer spans

3.1 Part-of-Speech Hierarchy

A part-of-speech (POS) is commonly defined as a linguistic category of lexical items that share some common syntactic or morphological characteristics. However, despite the concept of a POS being thousands of years old, grammarians and linguists still cannot agree on what exactly the shared characteristics are. This question has very real practical consequences: depending on where the line is drawn between common characteristics and distinguishing differences, one can end up with anywhere from the eight parts-of-speech defined in English textbooks to the 198 parts-of-speech of the London-Lund Corpus of Spoken English [10].

Our solution to this problem is to organize lexical items not into a single set of parts-of-speech but into a part-of-speech hierarchy. The lower levels of this hierarchy feature a greater number of finely-differentiated parts-of-speech, starting with the Penn Treebank POS tags [10] at the lowest level, while the higher levels contain fewer and more general parts-of-speech, and the topmost level is a single all-language-encompassing "universe" part-of-speech. Another innovation has been the inclusion in our hierarchy of several "blank" parts-of-speech, to represent and distinguish between the absence of different types of words. In total, our hierarchy contains 165 parts of speech organized in six levels. We originally developed it in the context of a keyword-extraction project; a detailed description of the hierarchy can be found in the paper describing that project [5].

We can define the semantic importance of different lexical items by assigning weights on the connections between POS in the hierarchy. This allows us to specialize the hierarchy for use in different NLP tasks. In our earlier work on keyword extraction [5], weight was given to verbs and nouns – the typical parts-of-speech of the keywords we were looking for. For question type classification, however, verbs and nouns are not the most semantically important words to take into account. Indeed, Tomuro [7] has shown that question type classification relies mostly on closed-class non-content words. The semantic weight in our hierarchy was thus shifted to the sub-hierarchies containing popular query adverbs and pronouns, including the wh-terms. The value of each POS in our hierarchy is then computed on the basis of its semantic weight and of the number of descendents it has. The resulting hierarchy, with the value of each POS, is presented in Figure 2.

We showed in [5] how using a part-of-speech hierarchy makes it possible to mathematically define several linguistic comparison operations. It is possible to compute the similarity between two words or POS simply as a function of their distance in the hierarchy. This computation takes several factors into account, including the number of levels in the hierarchy that need to be traversed on the path from one POS to the other and the value of each intermediate POS visited. Likewise, we can find a general place-holder POS to represent two words simply by finding the lowest common ancestor of both words in the hierarchy, and the similarity of the place-holder compared to the original words it represents is again a function of the distance in the hierarchy between each word and the place-holder POS. By extension, we can measure the similarity between two sentences by pairing their words together by similarity; the fact that our hierarchy includes "blank" parts-of-speech means that the two sentences do not need to be of the same length to be compared. And finally, we can merge two sentences by pairing their words together by similarity and replacing each pair by its lowest common ancestor in the hierarchy.

It is now straightforward to see how our part-of-speech hierarchy can be used for the task of question type classification. Given a list of informers representing different question types, we can use the hierarchy to compute the similarity between a user-specified query and each informer. The query is then classified to the same type as the informer it is most similar to.

3.2 Informer-Learning Algorithm

In addition to being used to compare queries and informers together, our part-of-speech hierarchy can be used as the core of an informer-learning algorithm. The intuition behind the learning algorithm is that, when merging two queries together, irrelevant words will be replaced by high-level POS while informer words common to both queries (if there are any) will remain words or low-level POS in the merged query. We can then extract the informer from the merged query by deleting POS of a level higher than a set threshold and keeping only contiguous words and POS below that threshold.

The principle underlying our learning algorithm is to start with an initial and incomplete set of general informers, and to learn more specialized informers to classify queries that cannot be correctly handled by the initial informers. The algorithm we use for this purpose is summarized in Figure 3. It takes as input a set of training

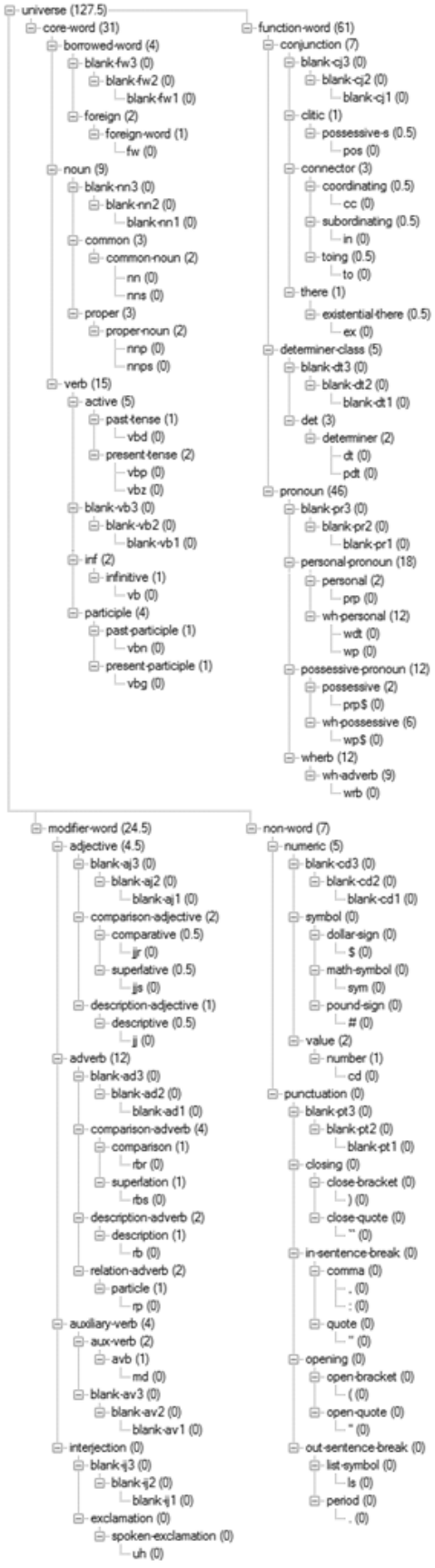


Figure 2: The POS hierarchy; POS values in parenthesis

queries classified in their correct types and a list of informers representing each type such as the list already proposed in Figure 1. It then divides the set of training queries into two subsets, one containing queries that can be correctly classified by the current informers and the other containing queries that cannot. In Figure 3, we call these sets C-queries and I-queries, for Correctly-classified queries and Incorrectly-classified queries respectively. The learning algorithm then merges together pairs of incorrectly-classified queries of the same type to generate new informers. A good informer that can be used to correctly classify some of the incorrectly-classified queries is added to the list of informers, and the queries it correctly classifies are moved to C-queries, the set of correctly-classified queries. This new informer might cause the misclassification of a few previously-correctly-classified queries though, and these queries are moved from C-queries to the set of I-queries to be used to learn more informers. The enriched list of informers is the final result of the learning algorithm.

```
1. Input: list of informers, training
queries
2. For each training query
    3. Classify using informers
    4. If classified correctly, add to
       C-queries
    5. Else, add to I-queries
6. For each pair of queries of same
type in I-queries:
    7. Merge and generate new informer.
    8. Classify all queries with new
       informer. Note which C-queries
       change to become misclassified,
       and which I-queries change to
       become correctly classified.
    9. If informer correctly classifies
       more I-queries that it
       misclassifies C-queries, add to
       the list of informers.
    10. Move correctly-classified I-
       queries to C-queries and
       misclassified C-queries to I-
       queries.
11. Return the list of informers
```

Figure 3: Structure of the learning algorithm

4 Experimental Results

For our experiments, we built a test corpus of queries using the 459 queries from the 2006 TREC QA track [11]. We tagged the words of the queries with their parts-of-speech using the standard Brill tagger, and we manually classified the queries into their correct question types.

Our first two experiments are designed to study the impact of using our POS hierarchy on the classification results. We do this by classifying the test queries using our

16 basic informers with and without our part-of-speech hierarchy. Classification using the hierarchy is done as described in Section 3.1. Classification without the hierarchy is done by simple keyword match; for example, any query that contains the word "who" followed by a present-tense verb will be classified in the person type. The results without using the hierarchy are meant to be a benchmark. They will show how well a system can perform the classification task by only recognizing the basic informers in the queries.

Next, we used the learning algorithm to expand the list of informers. The set of training queries we used is the query list from the 2007 TREC QA track [12], which we tagged and classified into correct types in the same way as we did for the 2006 TREC queries. The algorithm learned 11 new informers, which we present in Table 1. In the informers in that table, like in those in Figure 1, actual words are written plainly while POS from our hierarchy are written in square brackets. The words corresponding to the informers in the sample queries are marked in bold.

Table 1: New informers learned

Informer	Question type	Sample query
[vbd]	person (who)	**Name** members of the group.
what was	person (who)	**What was** his original name?
what [common-noun] [present-tense] [vbn][in]	person (who)	**What actor is used as** his voice?
[wh-personal]	location (where)	**Which** cities have Crip gangs?
[wh-personal] [present-tense]	location (where)	Name cities **that have** an Amtrak terminal.
what country	location (where)	With **what country** are they associated?
what countries	location (where)	**What countries** does it operate in?
[wh-personal] was	physical object (what)	**Which was** the first shuttle?
what [nn] of	physical object (what)	**What branch of** the military did he serve in?
what does	description (how/why)	**What does** the name mean or come from?
whan kind of	description (how/why)	**What kind of** insect is a boll weevil?

In each experiment, we computed the precision and recall of the classification of queries into each of our six question types, using the standard equations given in (1) and (2). We then computed the average precision and recall over all six types, and the average F-measure using equation (3). The results of these three experiments are given in Table 2.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \quad (1)$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \quad (2)$$

$$\text{F - Measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} \times \text{Recall}} \quad (3)$$

Table 2: Question type classification results

#	Experiment	Precision	Recall	F-Measure
1	Without hierarchy	64%	54%	58%
2	With hierarchy	90%	58%	70%
3	With learning	80%	65%	71%

4.1 Discussion

From the results presented in Table 2, it can clearly be seen that our system (line #2) performs much better than the benchmark (line #1). It achieves a 34% improvement in precision and a small improvement in recall. The reason for this improvement, and the only difference between our system and the benchmark, is the use of the part-of-speech hierarchy in addition to the 16 basic informers. The benchmark system can only exactly match queries to the informers; if there is no such match or if several informers appear in a query, the system is clueless. On the other hand, our system compares queries and informers together using the POS hierarchy as described in Section 3.1, and computes the similarity of each pair. In other words, it associates a query to its most similar informer, rather than look for an exact match. This ability to handle similar but inexact matches is clearly an important advantage.

In our third experiment, the learning algorithm discovers 11 new informers. It can be seen from Table 1 that they do represent different query syntaxes that were not accounted for in our initial 16 informers. The classification results using our hierarchy and the informer list including these new informers (line #3) show an important improvement compared to the benchmark (line #1). However, the advantage is less clear when compared to the 16 basic informers alone (line #2). The learning algorithm we outlined in Figure 3 is designed to accept new informers to correctly classify misclassified queries; in other words, to improve recall, which was low using the initial 16 rules alone. And we can see from the experimental results that our system's recall has indeed increased, but precision degraded, and one balances out the other so that the F-measure of our system remains almost unchanged.

Although both experiments using the POS hierarchy outperform the benchmark, there is still clearly room for improvement. Errors in our system are misclassifications caused by an informer from a wrong question type being more similar to a query than any of the informers from its own type. The main bad informer is "what [nn]", one of

our initial 16 basic informers. In the test using only those 16 informers, that one alone is responsible for 69% of the misclassifications our system makes. As we explained in Section 3.2, the idea behind our learning algorithm is to discover new informers to represent different query syntaxes and to correct classification mistakes caused by our set of initial informers. In practice, we find that four of the 11 new informers learned are inded used to correct misclassifications from the "what [nn]" rule, and in the new rule base the proportion of misclassifications due to that informer has dropped to 46%.

5 Conclusion and Future Works

In this paper, we present a new method to learn and apply informers for the task of syntactic-pattern-based question type classification. What sets our method apart is the use of our part-of-speech hierarchy, which makes it possible to compute mathematically the distance between the informers and the queries, and to associate queries with their most similar informers rather than look for exact keyword matches. Experimental results show that using the hierarchy for this task yields an average 12% improvement in the F-measure of type classification. Another advantage of our method is that it obtains good results using only 16 simple informers, while traditional methods use hundreds of keywords, patterns or probabilities.

Future work will focus on further improving the classification by refining the informer-learning algorithm. One thing that we noticed is that 8% of our training queries remained in the I-query set at the end of the learning algorithm. These are queries that could not be merged to any other to learn usable informers. Sometimes this is not because there are no similar queries to pair them with and learn from, but rather because there are too similar queries. For example, there are a lot of date-type queries that use the "what year" construct, but they also have other words in common near those two that are included in the informer extracted from the pairs of queries. Consequently, the words "what year" is never discovered as an informer by our system. A solution to this problem that we will next investigate is to learn informers from clusters of homogeneous queries instead of pairs of queries. The similarity measure between queries that we can compute using our POS hierarchy could be used to measure the homogeneousness of query clusters, and the informer learned for a cluster would be the result of merging all queries in a cluster together. This new clustering-based learning algorithm should be able to discover more precise informers than the current one.

6 References

[1] Majid Razmara, Andrew Fee and Leila Kosseim. "Concordia University at the TREC 2007 QA track"; Proceedings of the Sixteenth Text REtrieval Conference (TREC 2007), 2007.

[2] Noriko Tomuro. "Interrogative reformulation patterns and acquisition of question paraphrases"; Proceedings of the Second International Workshop on Paraphrasing, vol. 16, pp. 33-40, 2003.

[3] Cheng-Lung Sung, Min-Yuh Day, Hsu-Chun Yen, and Wen-Liar Hsu. "A template alignment algorithm for question classification"; IEEE International Conference on Intelligence and Security Informatics, pp.197-199, June 2008.

[4] Vijay Krishnan, Sujatha Das, and Soumen Chakrabarti. "Enhanced Answer Type Inference from Questions using Sequential Models"; Proceedings of Human Language Technology Conference / Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005), pp. 315-322, 2005.

[5] Richard Khoury, Fakhri Karray, and Mohamed Kamel. "Keyword extraction rules based on a part-of-speech hierarchy"; International Journal of Advanced Media and Communication, vol. 2 no. 2, pp. 138-153, 2008.

[6] Zhang Liang, Zhou Lang, and Chen Jia-Jun. "Structure analysis and computation-based Chinese question classification"; Sixth International Conference on Advanced Language Processing and Web Information Technology (ALPIT 2007), pp. 39-44, 2007.

[7] Noriko Tomuro. "Question terminology and representation of question type classification"; Second International Workshop on Computational Terminology, vol. 14, 2002.

[8] Sanda Harabagiu, A Harabagiu, Dan Moldovan, Marius Pasaca, Rada Mihalcea, Mihai Surdeanu, Razvan Bunescu, Roxana Girju, Vasile Rus, and Paul Morarescu. "Falcon: Boosting knowledge for answer engines"; Proceedings of the 9th Text REtrieval Conference (TREC-9), pp. 479-488, 2000.

[9] Dongsong Zhang, Jay F. Nunamaker. "A Natural language approach to content-based video indexing and retrieval for interactive e-learning"; IEEE Transactions on Multimedia, vol. 6, issue 3, pp. 479-488, 2000.

[10] Mitchell P. Marcus, Beatrice Santorini, Mary Ann Marcinkiewicz. "Building a large annotated corpus of English: the Penn Treebank"; Computational Linguistics, vol. 19, no. 2, pp. 313-330, 1993.

[11] Hoa Trang Dang, Jimmy Lin, and Diane Kelly. "Overview of the TREC 2006 Question Answering Track"; Proceedings of the Fifteenth Text REtrieval Conference (TREC 2006), 2006.

[12] Hoa Trang Dang, Diane Kelly, and Jimmy Lin. "Overview of the TREC 2007 Question Answering Track"; Proceedings of the Sixteenth Text REtrieval Conference (TREC 2007), 2007.

An Intelligent Othello Player Combining Machine Learning and Game Specific Heuristics

Kevin Cherry and Jianhua Chen
Department of Computer Science, Louisiana State University, Baton Rouge, Louisiana, U.S.A.

Abstract - *In this paper we present an intelligent Othello game player that combines game-specific heuristics with machine learning techniques for move selection. Five game specific heuristics have been proposed; some of which can be generalized to fit other games. For machine learning techniques, the normal Minimax algorithm along with a custom variation is used as a base. Genetic algorithms and neural networks are applied to learn the static evaluation function. The game specific techniques (or a subset of) are to be executed first and if no move is found, Minimax is performed. All techniques, and several subsets of them, have been tested against three deterministic agents, one non-deterministic agent, and three human players of varying skill levels. The results show that the combined Othello player performs better in general. We present the study results on the basis of performance (percentage of games won), speed, predictability of opponent, and usage situation.*

Keywords: Othello, genetic algorithm, neural network, minimax, influence map, expected min

1 Introduction

In Othello, two players, white and black, alternate turns in an attempt to capture as much of the 8x8 board as they can. Each valid move flips over one or more opponent pieces to display the player's color. At the end of the game, the player with the most cells showing his color wins.

The game starts out with two white and two black pieces as shown in figure 1 below. The black player always goes first and must choose from the valid moves highlighted in figure 2 to place one of his black pieces.

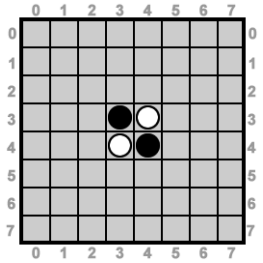


figure 1

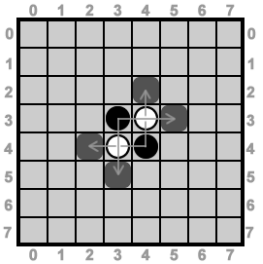


figure 2

Each one of black's valid locations result in a white piece being surrounded by two black pieces (one already on the board and the new one just placed). This means the surrounded white piece will be flipped over to its black-colored side and will become a black piece. For example, if the black player places a piece at row 5, column 3, the resulting board state would become:

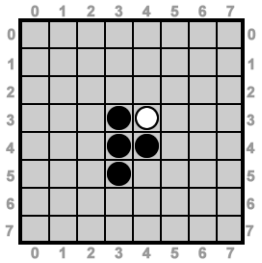


figure 3

Only moves which result in one or more opponent pieces being flipped are valid. At this point the white player would take a turn, with his valid moves being those that surround one or more black pieces either horizontally, vertically, or diagonally. Turns continue to alternate until either all board locations are occupied or one player has no more pieces left on the board. If a player does not have any valid moves, then that player's turn is forfeited.

2 Common methods

Along with game specific heuristics, which will be discussed later, an algorithm called Minimax was used. Minimax is an exhaustive search approach to finding an ideal move among the valid choices [1]. The player is known as "max" and the opponent is "min". To build the initial tree, the current board state starts as the root, then each of the player's valid moves become a child node, then each of the opponent's moves in response to the player's moves become children of that node, and so on. The tree construction stops when the specified depth has been reached. This basic structure is shown in figure 4 below.

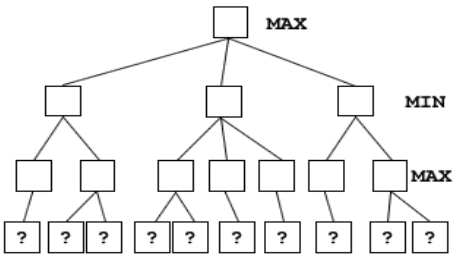


figure 4

These leaf nodes get their value from a static evaluation function. This function takes in features of a board state and assigns a real value indicating how good that state is for each player. Typically more negative values indicate good board states for min, while more positive values are good for max. For this paper a neural network was used for the function and certain input values as well as weights were trained using genetic algorithms. Figure 5 below shows an example of what the tree structure may look like after running the evaluation function. These leaf values must then be propagated upwards. The level right above the leaf nodes is a max level, meaning each of these nodes chose the maximum value from its children as that is better for the max player. The next level represents min's move and so the minimal value of each child node is chosen. The final level is always max since we are evaluating a move for the max player. After this root node gets a value, the path becomes clear and is highlighted below:

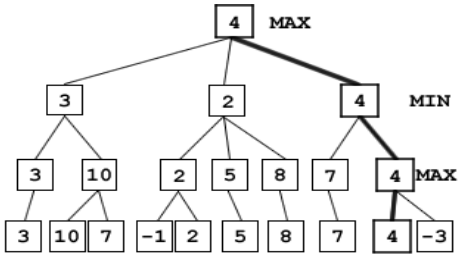


figure 5

This path represents a board configuration after a certain number of moves that is the result of the max and min player both playing optimally.

Normally the deeper the tree is allowed to build toward, the more accurate the node values. The number of nodes grows exponentially, however, resulting in an estimated total number of b^d, where d is the maximum depth and b is the "branching factor" or average number of children for each node [2]. This is why several optimizations were created for this algorithm. The one used for this paper is called alpha-beta and will be discussed later.

3 Game-specific heuristics

The game's rules are easy enough to learn, however there are several aspects that must be considered to become an expert player. First, the number of pieces one has during the beginning or middle game is not a good indication of how well one is doing [3]. Since single moves can often flip several pieces and change the game's score dramatically, it becomes more important to focus on getting one's pieces in certain key locations. These locations will set the player up for higher scoring moves later on. It is this reason that always choosing the move that results in the highest score gain is not an ideal strategy. Second, is about learning where these prime locations are and setting up moves that will help secure them. This also means ensuring one's opponent does not take these locations first. Lastly, one should try to look a few moves ahead to ensure that each move made will result in a more dominant board state later on.

Any good Othello agent must make use of these aspects in specific and general heuristic ways. There are several specific exploits designed into our agent that are tried first before relying on a more general approach. The order in which they are presented below is the order of precedence. Techniques are applied according to this precedence and if a move is found, that move is used and no further processing is done, else the next technique is used. The only exception to this comes with blacklisting, where moves are not chosen, but forbidden from being chosen by the sequential decision methods applied.

3.1 Killer move detection

The first exploit is killer move detection. If a player loses all his pieces on the board, that player has lost and the game is over. Therefore all valid moves are checked to see if any one move leads to the opponent having this outcome.

3.2 Corner detection

Second is corner detection. One set of prime locations are the corners of the board. This is due to the fact that a corner can never be overtaken once a piece is placed there and they allow for high scoring moves later on in the game. By always picking a corner location if it is in the list of valid moves, the agent can take advantage of these facts.

3.3 Blocking

Blocking is the third exploit and it attempts to keep the opponent from making a move. If any move in the valid move list forces the opponent to forfeit its next turn, that move is chosen. Even if this move is not an ideal choice, gaining an extra turn should make up for this.

3.4 Pattern detection

It is important to recognize board situations in which an ideal move is known. This sort of pattern detection with board states makes up the fourth exploit. The idea is to express several specific board states along with a collection, ordered by precedence, of predetermined ideal moves, or

locations to attempt to overtake with a single pattern. Several of these patterns can be created to cover a large number of board states.

The patterns created for the agent allow it to take a corner on its next move. This is accomplished by detecting a diagonal series of opponent pieces leading up to a corner and finding a way to overtake one of those pieces. This gives the agent a diagonal capture line to the corner on its next turn. Figure 6 below shows a possible board state that the mentioned pattern would recognize. The agent is in white and its valid moves are shown as dotted circles. The opponent pieces with white crosshairs are possible targets to overtake. The pattern does not recognize the opponent piece at (1, 1) since overtaking that spot would give the opponent the corner. Piece (4, 4) is also not a target since the opponent could flip that piece right back over on its next turn due to it having a piece at (3, 3). Since one of the agent's valid moves takes the target opponent at (2, 2), that move is chosen. The result of this move leaves the agent with a valid move at the corner, as shown in figure 7 below:

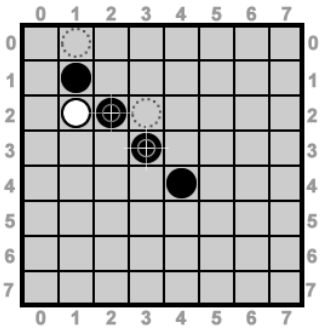


figure 6

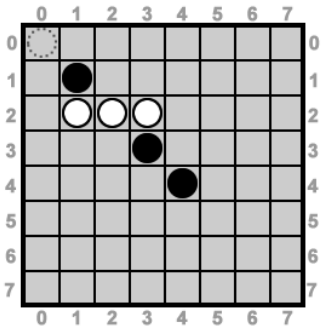


figure 7

Notice that the spot at (2, 2) cannot be taken back by the opponent's next move, as it is protected by the opponent's own pieces.

The implementation behind the other exploits are trivial enough to warrant not going into more detail, however construction of patterns can be quite complex. One must take a collection of several (possibly hundreds) of specific board states and represent those with a single structure along with actions to take if said pattern is matched. Therefore the implementation used will be discussed as one possible approach to creating this exploit.

A pattern was represented as an xml file that used bit masking to represent a location's status. One was untaken or empty, two was taken by the agent, four was taken by the opponent, and zero was inherited from another xml file called a template. Templates were used to store bit masked values for board locations common to multiple patterns (to keep from repeating the same value multiple times and make it easier to change if need be). For convenience if a location was not specified, it was assumed to have a "don't care" value and was ignored when checking the pattern. The xml file would then contain a list of rows and columns with bit masked states for each. For a pattern to match, each location, described by its row and column, would have to have a matching state to that of the actual board.

The pattern then specified which location it wanted to overtake. If this location was empty, the agent would simply place a piece there. If the location was taken by its opponent, it would look through its list of valid moves and choose one that would flip over that piece, thereby taking the location over. If no valid move could accomplish this, the pattern would move on to its next target location if one existed. If no target locations could be overtaken, the pattern would be considered unmatched and the next pattern was evaluated.

3.5 Blacklisting

The final exploit was created to prevent the opponent from having an ideal valid move. Blacklisting is the act of banning valid player moves that lead to this. These banned moves are then forbidden from being chosen by the next phase. If all moves are banned, then the banned list is cleared. It is important to not be too aggressive when banning moves as moves that seem to give the opponent the upper hand, might actually prove to be a better move for the player later in the game. Therefore banning moves too often can actually reduce performance.

4 Machine learning techniques

In this paper genetic algorithms were used to learn the influence map for the board, and both genetic algorithms and neural networks for the weights of the static evaluation function for Minimax. We also propose a variation on standard Minimax called "expected min" Minimax.

4.1 Expected min

If all previous game-specific exploits fail to find a move, then Minimax is run on all valid moves that are not blacklisted. In addition to the normal Minimax algorithm, a variation called "expected min" Minimax was added. This addition addresses a crucial issue with normal Minimax in its assumption over the actions of the min player. Instead of assuming the min player will follow the strategy of a subjective evaluation function, a more general approach is used to find an accurate representative value for each min node. The steps are as follows:

1) Take all child node values.
2) Subtract each value by the maximum of those values plus 1 (e.g. if we have 1, 2, and 3 then produce (1 – 4), (2 – 4), and (3 – 4) to get -3, -2, and -1). The reason for this is due to both the desire to end up with higher weights on lower numbers, and also to allow

values of zero to have some contribution to the weight distribution.

3) Sum these new values up and divide each value by that sum (e.g. for the -3, -2, and -1 values from above, we have (-3 / -6), (-2 / -6), (-1 / -6) to get 0.5, 0.333, 0.1667).

4) Multiply the original values by these weights (e.g. our original values of 1, 2, and 3 become (1 * 0.5), (2 * 0.333), and (3 * 0.1667) to get 0.5, 0.667, 0.5).

5) Sum these values up to get the min parent's value (e.g. 0.5 + 0.667 + 0.5 = 1.667).

This value is therefore taken from all child nodes with more weight added to lower values.

4.2 Static evaluation function

Minimax's static evaluation function takes the form of a neural network [4] with eight input nodes, one layer of five hidden nodes, and a single output node giving the final value. Sigmoid was used as the activation function [5] and each input value is evaluated for both the agent and its opponent. The input values are: the number of corner and side locations held, number of unique pieces that could be overtaken if the other player would have the next move, and the influence map sum of all player pieces on the board.

4.3 Influence map

An influence map is used to display the relative important of overtaking each location on the board [6]. These values should be learned to be effective, however with a range of zero to ten and 64 board locations, a search space of 11^{64} is too massive. Certain exploits and commonalities were therefore used to reduce this space. Assigning a value of ten to the corners and zero to the four starting locations, this space can be reduced to 11^{56}. Since one side of the board is no more important than the other, the influence map can be divided diagonally in both directions to form four quadrants with equal values. Each quadrant need only hold eight possibly unique values. After this the search space is reduced to 11^8 = 214,358,881. In the chart below, only values A to H need to be learned:

	0	1	2	3	4	5	6	7	
0	10	A	C	F	F	C	A	10	0
1	A	B	D	G	G	D	B	A	1
2	C	D	E	H	H	E	D	C	2
3	F	G	H	0	0	H	G	F	3
4	F	G	H	0	0	H	G	F	4
5	C	D	E	H	H	E	D	C	5
6	A	B	D	G	G	D	B	A	6
7	10	A	C	F	F	C	A	10	7
	0	1	2	3	4	5	6	7	

figure 8

The influence map values were learned using genetic algorithms [7] with the fitness function:

$$Fitness(W, W_r, L, L_r, S_w, S_l, N_c) = (W * W_r - L * L_r) + (w_{sw} * S_w - w_{sl} * S_l) + (w_{nc} * N_c) \quad (1)$$

Where:

$$W = \begin{cases} 1, & if\ agent\ won \\ 0, & otherwise \end{cases}$$

$$L = \begin{cases} 1, & if\ agent\ lost \\ 0, & otherwise \end{cases}$$

$$W_r = \begin{cases} \min\left({}^{agent\ pieces}/_{opp\ pieces},\ 5\right), & if\ opp\ has\ any\ pieces \\ 5, & otherwise \end{cases}$$

$$L_r = \begin{cases} \min\left({}^{opp\ pieces}/_{agent\ pieces},\ 5\right), & if\ agent\ has\ any\ pieces \\ 5, & otherwise \end{cases}$$

$$S_w = \begin{cases} 0, & if\ opp\ has\ any\ pieces \\ 1, & otherwise \end{cases}$$

$$S_l = \begin{cases} 0, & if\ agent\ has\ any\ pieces \\ 1, & otherwise \end{cases}$$

$$N_c = number\ of\ corners\ held\ by\ agent$$

The variables W_{sw}, W_{sl} and W_{nc} are the weights for S_w, S_l and N_c, respectively and were set at 5, 5, and 2, respectively. These values represent heuristic estimates. Each generation contained 21 chromosomes and selection, crossover and mutation operations were defined and used at a rate of 0.25, 0.75 and 0.05, respectively. Note that the mutation rate was chosen such that a single chromosome would be affected each generation. Chromosomes were chosen proportional to their fitness (i.e. chromosome fitness divided by total population fitness, or fitness proportionate selection [8]) and were moved onto the next generation or faced single-point crossover. Mutation was then applied by selecting random chromosomes at the before mentioned rate, and randomly changing one of its weights to a random value in the valid range of zero to ten. Figure 9 shows how the fitness changed with each generation during the learning phase:

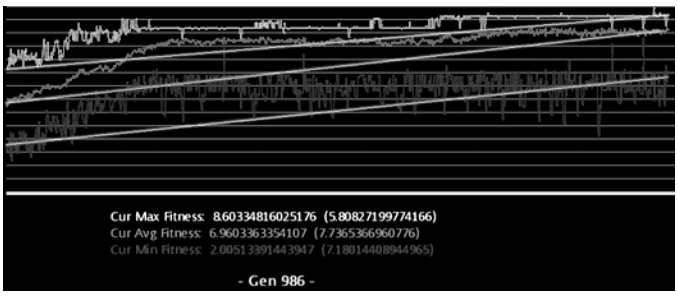


figure 9

In figure 9 "Gen" is the current generation and the white, light gray, and dark gray lines indicates max, average, and min fitness respectively. There is a white line for each of these that shows the progress from the initial generation. To get the fitness value of a chromosome, that chromosome's knowledge was put into an influence map agent (an agent who only uses its influence map to make its decision) and that agent played against the target agent as the white and black player. The fitness of both games was averaged to get that chromosome's actual fitness. This was run initially with a greedy agent (one whose move choice is based solely on the number of opponent's pieces it can overtake) as the target, however after a fitness of eighteen was reached, the most fit chromosome's knowledge was put into an influence map agent to be the new target agent. The genetic algorithm was then started over with all chromosomes testing their knowledge against the target agent. This repetition continued for approximately six times before the final fittest chromosome's knowledge was used. Its influence map was then used in the input to the static evaluation function.

4.4 Training the network

With all inputs defined and learned, the neural network must be trained. Genetic algorithms were again used for training. With eight input, five hidden, and one output node, the number of weights, along with the bias nodes, is 51. Population size was increased to 100 to deal with this larger space and mutation rate decreased. The new mutation rate of 0.01 was chosen such that only one chromosome would be affected; same as last time. Selection and crossover rates as well as fitness function remain the same. Each chromosome's knowledge was put into a Minimax agent and used for its evaluation function. This agent played as white and black player against the greedy, influence map (using the same map), and greedy influence (combination of two approaches) agents. A total of six games were therefore played for each chromosome and the fitness was averaged. The results of running the genetic algorithm on four different machines in parallel, each with their own initial population, for around 80 hours is shown in table 1. Notice the best chromosome had a maximum fitness of 14.22, which is quite good.

table 1

Number of generations	14,608	11,769	11,808	12,597
Maximum fitness obtained	8.95	14.21	14.22	11.24

To speed up the process during training, Minimax used a max depth of one. Since it is only necessary to find neural network weights that can accurately describe the board state given, a depth of one was enough to accomplish that and any higher depth would have added unnecessary training time [9]. Also to aid in training, the mutation rate was sometimes adjusted when the genetic algorithm appeared to hit a plateau. This occurs when generations seem to all have similar fitness values as nothing new is being added between populations. This could indicate a local minimum and increasing the mutation rate is a way of introducing added variation to help overcome this.

To get a better idea of how the Minimax depth affects performance, maximum depths of one to six were used for testing. The max of six was used based on the amount of time taken to play a single game and personal thoughts on how many depths are sufficient enough for testing.

Alpha-beta is among the most common Minimax optimizations [10]. It works by figuring out which nodes do not contribute to the final root value. For example, say we have two min level nodes, A and B, with A having a value of four and B unevaluated. B's children would all have to have values greater than four for B to be selected, since B chooses the minimum value of its children. Therefore after evaluating each child node, if the value is ever less than four we can stop. At this point we know that B's value will be less than four and A will be chosen. Therefore the final tree might look something like this:

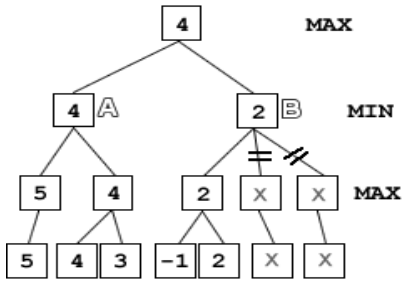


figure 10

The nodes with X's were not evaluated and their subtrees were pruned. This procedure can be performed at any level to eliminate unnecessary node evaluation and cause the entire Minimax algorithm to be a lot less expensive.

Although alpha-beta was used to help speed up testing, it could only be used on normal Minimax. The expected min algorithm discussed earlier requires all child nodes to form an accurate value for the min parent, and therefore cannot or should not be combined with any pruning optimizations.

5 Experiments

Testing was performed against the three deterministic agents mentioned previously, a random agent and three human players. The greedy agent always takes the move that results in the most opponent pieces being flipped. The influence map agent chooses the move with the highest corresponding value from our influence map. Greedy influence multiplies the number of opponent pieces flipped with the influence map value for each move and picks the one with the highest result. Finally the random agent selects a valid move at random. The human players had self-rated skill

levels of one, five, and seven with one being a beginner and ten a very experienced player.

5.1 Artificial agents

Here are the results of the agent playing against the three deterministic agents:

tables 2 (top) and 3 (bottom)

Without EM								
	M	**MCK**	**MCKBoBa**	**MCKBoP**	**ALL**	**Total**	**Avg**	**Per**
1	6	6	5	5	5	27.0	5.40	90%
2	4	4.5	4	4	3	19.5	3.90	65%
3	4	4	4.5	4	4.5	21.0	4.20	70%
4	3	4	6	5	6	24.0	4.80	80%
5	4	4	6	5	6	25.0	5.00	83%
6	5	5	3	6	4	23.0	4.60	77%
Total	26.0	27.5	28.5	29.0	28.5			
Avg	4.33	4.58	4.75	4.83	4.75			
Per	72%	76%	79%	81%	79%			

With EM								
	M	**MCK**	**MCKBoBa**	**MCKBoP**	**ALL**	**Total**	**Avg**	**Per**
1	6	6	5	5	5	27.0	5.40	90%
2	1.5	2	4	1	4	12.5	2.50	42%
3	3	3	5	3	5	19.0	3.80	63%
4	3	5	6	4	5	23.0	4.60	77%
5	6	5	6	6	6	29.0	5.80	97%
6	4	5	4.5	5	4.5	23.0	4.60	77%
Total	23.5	26.0	30.5	24.0	29.5			
Avg	3.92	4.33	5.08	4.00	4.92			
Per	65%	72%	85%	67%	82%			

EM refers to the expected min approach. Each cell represents how many games out of six were won (1 game against each of the three agents as both players. Ties count as half a point). For the abbreviations under the table heading: 'M' is Minimax, 'C' is corner detection, 'K' is killer move detection, 'Bo' is blocking, 'Ba' is blacklisting, 'P' is pattern detection, and 'ALL' is for all techniques. Combinations indicate techniques that were used for those tests. The number in the first column represents the maximum Minimax depth searched to. 'Total' was the total number of games won (the sum of the appropriate cells), 'Avg' is the average, and 'Per' is the percent won. Finally, the cells shaded in light gray represent situations were all six games were won, cells in dark gray mean half or less of the games were won, and cells in mid gray represent the best Minimax depth or best combination of techniques for that chart. Even though the expected min approach performed worse overall, it had higher maximum scores with 97% over 90% for depth and 85% over 81% for technique combinations. This indicates some promise for the approach. Note that for expected min, depths one to five were used without alpha-beta pruning, however, due to memory constraints, it was applied for depth six. This caused the performance of depth six to be less than that of the true value. For normal Minimax this anomaly is also present and can be explained by examining the different exploit combinations. Notice that for those technique sets that do not include blacklisting, the performance actually improved from depth five. This would indicate that blacklisting does not perform well at this depth and is evidence of the precaution mentioned when blacklisting was introduced earlier. Basically blacklisting is being used in situations where Minimax produces better results. Therefore blacklisting is preventing what it feels is a bad move, when Minimax, having examined the possible board states 6 plies ahead, knows better. This leads to the point that, although each of these exploits can be used to improve performance, they all have the ability to decrease performance under certain circumstances. Therefore caution should be taken when choosing the correct combination (more on this later). Also, note that depth one did considerably better than most other depths in both situations. This is most likely the side effect of training at that depth. Another side effect was caused by the fact that these testing agents were the same as the training agents. Although the depths and techniques used differ, this fact could have caused distortion of the test results. It is therefore imperative that tests are run against unseen agents. Therefore the same tests were repeated against the random agent with the difference that each game was played ten times and scores were averaged. The agent's scores in every category were superior to that of the random agent.

5.2 Human agents

Next, human players tested the aptitude of the agent. Here are the results:

tables 4 (top) and 5 (bottom)

Without EM					
	M	**ALL**	**Total**	**Avg**	**Per**
1	3	5	8.0	4.00	67%
6	6	6	12.0	6.00	100%
Total	9.0	11.0			
Avg	4.50	5.50			
Per	75%	92%			

With EM					
	M	**ALL**	**Total**	**Avg**	**Per**
1	3	5	8.0	4.00	67%
6	3	6	9.0	4.50	75%
Total	6.0	11.0			
Avg	3.00	5.50			
Per	50%	92%			

Note that with human players, the amount of combinations, and therefore amount of games required, must be lowered in order to seem reasonable. The charts are in the same format as before and again alpha-beta was used with expected min at a depth of six. The results improved when the game-specific exploits were added as indicated by the "all" column and every exploit and depth category saw 50%

or higher cumulative scores (meaning the agent matched or exceeded the expertise of the human player).

5.3 Choosing the right parameters

When choosing the right exploits, Minimax depth, and whether or not to use expected min, one must consider the application of the Othello agent. Aspects to consider are that of move accuracy, time taken, and Minimax nodes examined (for memory constraints) and are summarized in the following figures and tables.

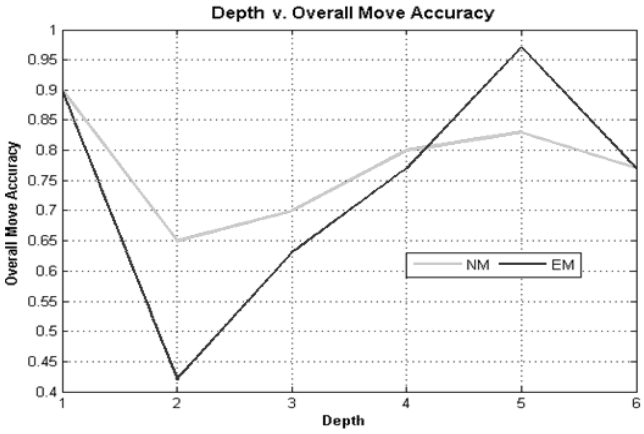


figure 11

Figure 11 shows the percent accuracy of move decisions for both expected and normal Minimax and how it changes with maximum depth. The data for this figure was taken from tables 2 and 3 above.

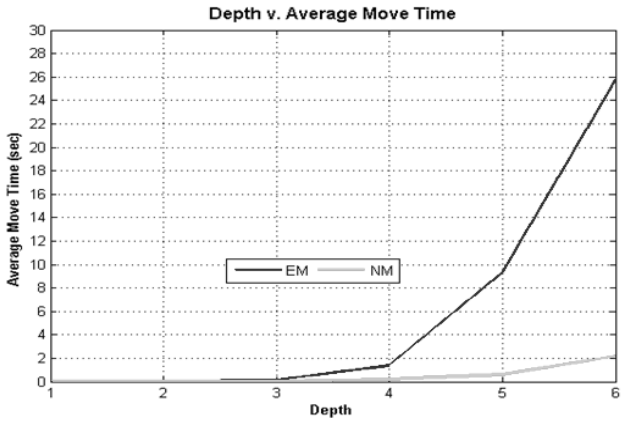


figure 12

table 6

Average Move Times (sec)						
	1	**2**	**3**	**4**	**5**	**6**
NM	0.002	0.008	0.063	0.197	0.609	2.179
EM	0.002	0.019	0.151	1.396	9.293	25.749

Figure 12 shows the information expressed in table 6 below it. In both, "NM" stands for normal Minimax and "EM" stands for expected min Minimax. Each depth can significantly increase the average amount of time (in seconds) taken for a move decision to be made. Since using expected min at depth six was not actually possible (due to memory constraints), the data for it was estimated using a third order polynomial with the data from depths one to five. The data for average number of nodes searched is highly correlated with average move time as it is the main factor in determining it. Table 7 shows this with the expected min value at depth six being estimated using the same procedure as in table 6.

table 7

Average Nodes Searched						
	1	**2**	**3**	**4**	**5**	**6**
NM	7	28	219	714	2,037	7,548
EM	7	66	497	4,502	28,862	79,496

If the agent application was that of a commercial game, move time would need to be minimized, nodes searched must be kept reasonable, and accuracy could be negotiable by the human player. For an easy difficulty setting, one might choose Minimax only with expected min and search depth of two. Normal difficulty could be Minimax with normal min searching to a depth of three with corner and killer move detection. For hard, normal min Minimax at depth six along with all exploits active. If the application was more of a test against other computer agents, move accuracy may be more important than quicker move times and nodes searched may not be of too much concern. If that is the case, a combination of expected min Minimax searching to a depth of five with corner detection, killer move detection, blocking and blacklisting would be a good choice.

6 Conclusion

This paper has shown the approaches used to create an intelligent Othello player. Game specific heuristics, namely killer move detection, corner detection, blocking, pattern detection, and blacklisting were created to help the agent choose an ideal move among the valid alternatives. If these fail to produce a decision, Minimax is ran using a custom static evaluation function learned by genetic algorithms and driven by a neural network. An alternative to the standard way of choosing a value for the min node was also presented. All of the above were then tested under the same situations and results were summarized and explained. Advice was given on selecting one's own combination of techniques for specific tasks. Overall the agent with all its methods performed quite well on all tests presented to it.

6.1 Future work

To improve upon the agent, some further work could be done. If one were to use a variation of the leave-one-out cross

validation by training on two of the three deterministic agents then testing on the third, and repeating this for each agent, a better approximation of the true performance could be obtained for the neural network static evaluation function. The neural network might also benefit from the addition of more input features. One such value might be the number of "strong" lines on the board for each player where strong is defined as a collection of pieces that are hard to overtake in few moves. For example if a player had pieces that formed a line from one side of the board to the other, there would be no way the opponent could overtake all pieces in a single move, and would probably require several instead. Also, the way in which the neural network learns could be changed to a reinforcement learning method as opposed to genetic algorithms as they might yield better weights. If one wanted to use this agent in an Othello game meant for human players, one could benefit from adding more Minimax optimizations such as Negascout [11] or other forms of aspiration search. Remembering ideal moves from past games and forming a history table may help to improve move times as more games are played. This could also lead to deeper Minimax searching by recalling past values found from evaluating the same board state. More patterns could help to capture these ideal moves, however they must be well thought out since, like blacklisting, they can be overused and block better decisions from Minimax.

7 References

[1] "AI Horizon: Minimax Game Tree Programming, Part 1." AI Horizon: Computer Science and Artificial Intelligence Programming Resources. AI Horizon. 27 February 2011. <http://www.aihorizon.com/essays/basiccs/trees/minimax.htm >.

[2] Rajiv Bakulesh Shah, "minimax," in Dictionary of Algorithms and Data Structures [online], Paul E. Black, ed, U.S. National Institute of Standards and Technology. 10 January 2007. 27 February 2011. <http://xlinux.nist.gov/dads//HTML/minimax.html>

[3] Rose, Brian. Othello: A Minute to Learn... A Lifetime to Master. Anjar Co., 2005.

[4] Graupe, Daniel. Principles of Artificial Neural Networks. Singapore: World Scientific, 1997. 1-3.

[5] Shepherd, Adrian J. Second-Order Methods for Neural Networks. Great Britain: Springer, 1997. 3.

[6] Matthews, James. "generation5 – Simple Board Game AI." generation5 – At the forefront of Artificial Intelligence. 27 December 2004. Generation5. 27 February 2011. <http://www.generation5.org/content/2000/boardai.asp>.

[7] Mitchell, Tom M. Machine Learning. Singapore: Mcgraw Hill, 1997. 249-270.

[8] Mitchell, Tom M. Machine Learning. Singapore: Mcgraw Hill, 1997. 255.

[9] Smed, Jouni, and Harri Hakonen. Algorithms and Networking for Computer Games. Finland: John Wiley & Sons, Ltd, 2006. 80-82.

[10] Shapiro, Stuart C. Encyclopedia of Artificial Intelligence. USA: Wiley-Interscience, 1987. 4-7.

[11] Reinefeld, Alexander. Die Entwicklung der Spielprogrammierung: Von John von Neumann bis zu den hochparallelen Schachmaschinen. Zuse Institut Berlin: Humboldt-Universität zu Berlin, 2005. 47-51.

Occam Learning through Pattern Discovery: Computational Mechanics in AI Systems

Dr. James A. Crowder, Dr. John N. Carbone
Raytheon Intelligence and Information Systems
JACrowder@raytheon.com, John_N_Carbone@raytheon.com

Abstract - *The push for real-time autonomous AI systems has been sought for decades. The DoD has spent considerable R&D budgets looking for systems that can operate with no or little supervision. These systems must process incredible amounts of heterogeneous information looking for information. In order to achieve these goals, we must affect real learning, or "learning with experience," in autonomous AI systems [10]. The goal of having machines that learn with experience is one of the most intriguing problems in computer science and computer engineering. As the types of problems we would like AI systems to solve get more complex and more diverse, it is becoming a necessary task as well. Unfortunately, by its nature, learning is somewhat fuzzy, and random in nature, for information comes at us in stochastic fashion [22]. In fact, the overall goal is to learn things we do not yet know, and in doing so find patterns that we can learn. This constitutes not patter matching, or pattern recognition, but is, in fact, pattern discovery. Nonetheless, we would like a mathematical framework for machine learning to aid in our understanding and improve our ability to make progress toward autonomous AI systems.*

1 Introduction

Multi-layer neural networks have been successfully applied to complex pattern recognition and functional approximation problems. The notion of pattern recognition is readily known and understood [3]. However, here we introduce a different concept in AI machine learning, the concepts of *Pattern Discovery*, and a different way of looking at learning for autonomous systems, whose input will come in very stochastic and unpredicted ways. Here we look at the concepts of finding causal structure in stochastic data that come out of computational physics.

The term Pattern Discovery is intended to be in contrast to the well understood concepts of both *Pattern Recognition* and *Pattern Learning*. In Patter Recognition the aim is to analyze input data and assign it to one or more pre-determined categories or patterns. In most Pattern Learning systems, the goal is to determine which of the several pre-determined categories available to the algorithms corresponds to the correct one [25]. Obviously the two are closely correlated and in both cases, the categories or representations have been handed to the algorithms, due to choices and decisions which are external or outside the pattern recognition and learning algorithms or procedures.

For our use of Pattern Discovery, the goal is to avoid the necessity for *a priori* knowledge about what structures, or patterns, may be relevant [20]. This is not a new problem, and is as old as the first attempt to process information. The classical approach, based on statistical mechanics, is to derive patterns (or macroscopic properties) from raw data (or microscopic components). Here we take the inverse approach, extending the concept of extracting "geometry" or causal structures, from a time or frequency series of data or information. Here we build upon the concept of "Occam Learning" [7] to construct the simplest model capable of capturing causal structures, or "patterns" in the data which constitutes a representation of the causal structure of the hidden process(es) which generated the behavior observed or captured. We assert that this representation is the maximally efficient model of the observed data-generating process, based on the learning principles laid out in the Occam Learning Process. The underlying computational mechanics concepts have been used to analyze dynamical systems, evolving spatial computation, and stochastic resonance, among others. However, the combination of computational mechanics coupled with the Occam AI learning constructs provides a unique method for Pattern Discover in large, heterogeneous data sets. These methods, combined with extensive AI cognitive processes, e.g., the Artificial

Cognitive Neural Framework (ACNF) [1, 4, 5, 6], move us toward providing real-time, completely autonomous AI systems.

Presented here is a mathematical framework for discovering, describing, and quantifying new patterns, based in computational mechanics and using tools from statistical physics. It construct optimal, minimal models of stochastic processes and their underlying causal structures that drive an "Occam Learning" model of intrinsic computational information transformation. Here we summarize the mathematical foundation of computational mechanics, especially those constructs in optimality and uniqueness to drive the Occam learning algorithms. We describe the principles and motivations underlying the computational mechanics, emphasizing its underlying connections to the minimum description length principles underlying Occam Learning, and its implications to Probably, Approximately Correct (PAC) machine learning concepts [2].

We will first examine the concepts and issues involving Pattern Discover and how they are addressed with computational mechanics. Then we provide discussion of the mathematical structures, based in computational mechanics that provide the foundation for Pattern Discovery, with particular attention to optimality and uniqueness theorems. These uniqueness theorems are then utilized within the Occam Learning framework to provide AI learning algorithms to learn and extend these unique pattern structures. Differences between this work and other work in computational mechanics is that we utilize Renyi's Entropy theory vs. Shannon's, utilizing Renyi's mutual information theory in our computations involving stochastic processes [6, 11].

2 Pattern Discovery Concepts

Any approach we take to handling Pattern Discovery should meet a number of criteria:

- ***Predictive*** – the models the algorithms produce should allow the system to predict the original process or system that produced the data, and provide a compressed description of it (learned pattern).
- ***Computational*** – should have stored in system memories how that process or system stores, transmits, and transforms information (what was the causal structure that produced the information?).
- ***Calculable*** – either analytically or by systematic approximation.
- ***Causal*** – system should understand how instances of the discovered patterns are actually produced.
- ***Naturally Stochastic*** – the learned patterns models should not just be tolerant of noise, but should be explicitly formulated in terms of stochastic ensembles.

For our uses, the key idea of utilizing computational mechanics is the supposition that the information required to drive the Occam Learning Pattern Discovery is actuall in the data, or information picked up by the system's sensors, provided there is enough information. The real problem dealing with real-time information coming from a number of possibly heterogeneous sensors is to determine which data sets (or partitions) of data sets should be treated as equivalent, and how the data should be partitioned. We assert that correct mapping of data to partitions should leave the system with discovered patterns that provide the system's cognitive framework with the same degree of knowledge about the future of the data (similar prediction accuracy and consistency), has a proper degree of plausibility, but is also vague enough to account for future growth of the learned pattern definition [17, 18]. So the question is: how to create the partitions? For this we look to genetic algorithms. We create generations of partitions and then use these to create Occam patterns, or memories, from the populations, based on the partition constraints (based on computational mechanics). These are evaluated, based on Entropy calculations. Those partitions from the population that produce the best fit and utilized (with mutation and crossover) are used to create a new generation of partition population. The process continues until an optimal partition is created. Those partitions that produce patterns similar to those already in the system's memories are sent to algorithms that evaluate the patterns for memory extensions or reinterpretations. Those that are not already part of the system's memories are used to create new memories.

For this discussion, we refer to ***H[X]*** as the entropy of discrete random variable ***X***, interpreted as the uncertainty in ***X***. ***H[X\|Y]*** is the entropy of ***X*** conditional on ***Y***, and ***I[X\|Y]*** is the mutual information between ***X*** and ***Y***, as measured by Renyi's Entropy and Mutual Information computations. Also, we restrict ourselves to discrete-valued, discrete-time stochastic processes (analogous to sensor data being collected by an autonomous system). Such processes are sequences of random variables, $\boldsymbol{S_i}$, and the values are taken from a countable set ***A***. This is reasonable since we are talking about a system with multiple sensors, each taking in data over a specified period of time, each with a countable number of data samples.

Our goal is to discover a pattern, or Occam memory that will predict all or part of the future of process $\boldsymbol{S}, \vec{S}$, using some function and some part of $\vec{S}$. We begin by taking the set $\vec{S}$ of past data points and partitioning it into mutually exclusive and jointly comprehensive subsets, as shown in Figure 1. That is, we make a class $\mathfrak{R}$ of subsets.

Patterns in Data Ensembles: In order to discuss Pattern Discovery in data ensembles, we must have a way to discuss the uncertainty of Occam memories to predict future information states.

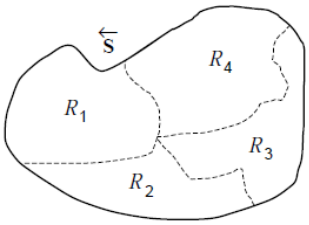


Figure 1 - $\mathfrak{R}$ Partitions of the set $\overleftarrow{S}$

We cannot use:

$H\left[\vec{S}\right]$ Since this is infinite. Instead we use:

$H\left[\overrightarrow{S}^{L}\right]$ where the uncertainty of the next ***L*** data is treated as a function of ***L***. Therefore, $\mathfrak{R}$ captures or discovers a pattern, ***iff*** there exists an L such that:

$$H\left[\overrightarrow{S}^{L} \mid \mathfrak{R}\right] < LH[S]$$

$\mathfrak{R}$ discovers a pattern when it tells us something about how the distinguishable parts of the process affect each other, or how $\mathfrak{R}$ exhibits its independence, based on the entropy calculations discussed earlier. The smaller that

$$H\left[\overrightarrow{S}^{L} \mid \mathfrak{R}\right]$$

is, the stronger the pattern discovered by $\mathfrak{R}$. The causal state, as determined by the Occam memory of a captured pattern, together with the next observed process, determine a new causal state (and may cause a redefinition of the Occam memory). Thus, there is a natural relation of succession among the causal states of a captured pattern or causal process. This leads us to the definition of a captured or discovered pattern, which leads to an Occam Memory within the AI system. Each discovered pattern, or Occam Memory will have the following properties:

- Occam Memories are deterministic
- All Occam Memory causal states are independent
- All Occam memories are reconstructed from information fragements
- All Occam memory causal states are maximally prescient
- All Occam memory causal states are minimal for all prescient rival memories
- All Occam memory causal states are unique
- All Occam memories are minimally stochastic for all prescient rival memories.
- The excess entropy, ***E***, of an Occam Memory is the Mutual Information between the memory's semi-infinite past and its semi-infinite future.

3 Computational Mechanics and Occam Learning

"*Entities should not be multiplied unnecessarily.*"
William of Occam (1320 AD)

This maxim from William of Occam, called "***Occam's Razor***," is often sited to justify one hypothesis over others, and is taken to mean "*prefer simpler explanations.*" However, what reason might we have to believe that simpler explanations lead us to a hypothesis with fewer errors?

One might simply reason this from the observation that there are far fewer simple explanations than complex ones. However, it may be no more complex than the reasoning that simple explanations are less likely to fit data, just by chance. Another way to view this is that by favoring smaller hypotheses over larger, we are less likely to run across bad hypotheses, which one of the fundamental axioms behind Occam Learning. Another axiom of Occam Learning is:

"***Learning is Data Compression***"

The more the data is compressed, i.e., the more complex the learning algorithm, the more likely something subtle that is important is missed or eliminated. Reasoning from this perspective, we define an "Occam Learning Algorithm" to be one that produces hypotheses, or "Pattern Discoveries, that are simple in structure, and grow slowly as more data are analyzed. In fact, analysis has shown [4, 6, 7] that if we have a small hypothesis space, then by taking a polynomial number of data samples, we can achieve "Uniform Convergence," i.e., the chance that any bad hypothesis with error > c, that is still consistent with the data, can be forced below some arbitrary number δ [23]. In the converse, is it impossible to get uniform convergence with a large hypothesis spaces, given a polynomial number of data samples, the answer is, sometimes [7].

Since learning is very stochastic in nature, particularly for real-time systems with heterogeneous data inputs, and given that it is impossible to know how many data points for a given unknown pattern may exist, we employ Occam Learning to provide Pattern Discovery. What we desire, then, is a mathematical framework and foundation for AI system learning, based in computational mechanics and Occam Learning principles to provide the AI system with autonomous understanding, reasoning, and decision.

Towards this end, a memory computational framework that encompasses the computational theory of machine learning discussed here, the goals of which are:

- To provide computational mechanics mathematical models that capture key aspects of Occam Learning.
- To provide the system self-analytical metrics for its algorithms:
  - When will they succeed?
  - How long will they take?
- To develop algorithms that provably meet desired criteria;
- To provide the system self-guidance about which algorithms to use when.
- To allow the system to analyze the inherent ease or difficulty of learning problems.

Figure 2 illustrates the Occam Learning, computational mechanics framework.

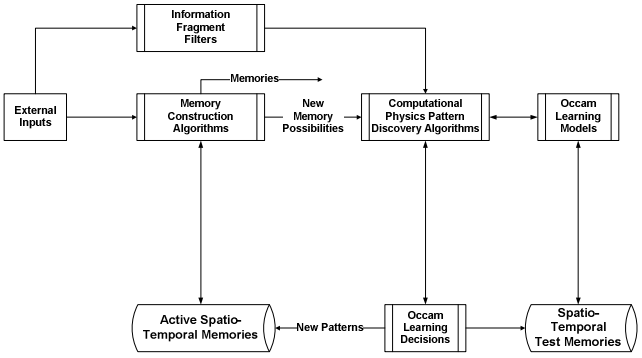


Figure 2 – The Occam Learning, Computational Mechanics Learning Framework

This Occam Learning system is one aspect of the "Learning Algorithms" depicted in the overall Artificial Cognitive Neural Framework depicted in Figure 3 [11]. The three main subsystems within the architecture are [10]:

- The Mediator: this gathers information and facilitates communication between software agents within the system [8]. The Mediator takes information from perceptrons and from coalitions of perceptrons and updates the short-term, long-term, and episodic memories.
- the Memory System: The information available in memory, i.e., what the system has learned, is continually broadcast to the conscious perceptrons that form the cognitive center of the system.
- Cognitive System: this is responsible for the cognitive functionality of perception [12], consciousness, emotions, information processing, etc.

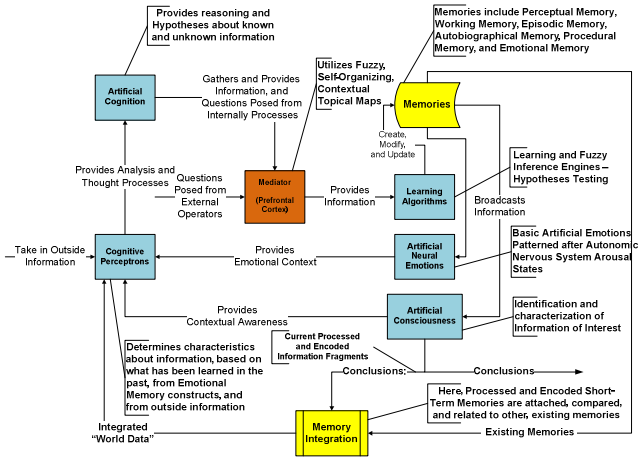


Figure 3 – The Artificial Cognitive Neural Framework

The Learning Center, or Learning Algorithms as it is depicted in Figure 3, provides a number of learning systems, of which Occal Learning is the most basic and fundamental. Other learning structures, such as the Probably, Approximately Correct (PAC) learning algorithms take their cues from the Occam Learning structures (patterns), and use combinations of these to infer higher-order learning within the AI system. These higher-order learning algorithms allow the system to "infer" new and more complex concepts, that turn into new, more complex memories, than it previously stored or had available. In this way, the system "evolves." The ACNF provides:

- An architectural framework for "conscious" software agents.
- A plug-in framework for the domain-independent portions of the "consciousness" mechanisms.
- An easily self-customizable framework for the domain-specific portions of the "consciousness" mechanisms.
- The cognitive mechanisms for behaviors and emotions required for the conscious software agents.

The ACNF contains several different memory systems (including emotional memories) each with a specific purpose. The Occam and PAC memory systems reside in both the Perceptual and Emotional memory systems [9]. The memory systems and their descriptions are [13, 14, 15, 16]:

- **Perceptual Memory**: this memory enables identification, recognition, and characterization, including emotions.
- **Working Memory**: this contains preconscious buffers as a temporary workspace for the internal AI system activities.

- **Episodic Memory**: this is a content-addressable associative memory with a rapid decay (very short-term memory) [2].
- **Autobiographical Memory**: this is the long-term bi-directional associative memory for facts and data.
- **Procedural Memory**: this is the long-term memory for learned skills within the system.
- **Emotional Memory**: both long-term (spatio-temporal) and implicit (inference) emotional memories.

4 Conclusions and Discussion

Here we have laid the foundations for learning structures that will be required for real-time autonomous AI systems. We have provided a mathematical basis for these learning algorithms, based in computational mechanics. The Occam Learning system is but one of many learning constructs that must be present for an AI system to actually act autonomously and to make sense of a complex world it will find itself a part of. The Occam Learning Computational Framework provides the ability for simple Pattern Discovery that feeds more complex memory and inference systems within the ACNF to allow the autonomous system to think, reason, and evolve. We have but scratched the surface in providing constructs and methodologies required for a self-aware, thinking, reasoning, and fully autonomous real-time AI system [19].

5 References

1. Crowder, J., Barth, T., and Rouch, R., "*Learning Algorithms for Stochastically Driven Fuzzy, Genetic Neural Networks*." NSA Technical Paper, Denver (1999) ENIGMA_1999_002.
2. Crowder, J., Barth, T., and Rouch, R., "*Neural Associative Memory for Storing Complex Memory Patterns*." NSA Technical Paper, Denver (1999) ENIGMA_1999_003.
3. Crowder, J., Barth, T., and Rouch, R., "*Evolutionary Neural Infrastructure with Genetic Memory Algorithms: ENIGMA Theory Development*." NSA Technical Paper, Denver (1999) ENIGMA_1999_004.
4. Crowder, J. A., "*Integrating an Expert System into a Neural Network with Genetic Programming for Process Planning*." NSA Technical Paper TIT_01_01_013_2001_001 (2001).
5. Crowder, J. A. "*ELYSE, The Evolving, Life-like Yielding, Symbiotic Environment."* NSA Technical Paper Technical Paper TIT_01_01_013_2001_002 (2001).
6. Crowder, J. A., "*Adaptive Learning Algorithms for Functional Constraints on an Evolving Neural Network System*." NSA Technical Paper CON_0013_2002_003 (2002).
7. Crowder, J. A., "*Machine Learning: Intuition (Concept) Learning in Hybrid Genetic/Fuzzy/Neural Systems*." NSA Technical Paper CON_0013_2003_009 (2003).
8. Crowder, J. A., "*Operative Information Software Agents (OISA) for Intelligence Processing*," AIAA Aerospace@Infotech 2010 Conference (May, 2010).
9. Crowder, J. A., Friess, S., "*Artificial Neural Emotions and Emotional Memory*." International Conference on Artificial Intelligence, ICAI'10 (July, 2010).
10. Crowder, J. A., "*Flexible Object Architectures for Hybrid Neural Processing Systems*," International Conference on Artificial Intelligence, ICAI'10 (July, 2010).
11. Crowder, J. A., Carbone, J. N., "*The Great Migration: Information to Knowledge using Cognition-Based Frameworks*." Springer Science, New York (2011).
12. Marsella, S., and Gratch J., "*A Step Towards Irrationality: Using Emotion to Change Belief*." 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems, Bologna, Italy (July 2002).
13. Newell, A., "*Unified Theories of Cognition*." Cambridge MA: Harvard University Press (2003).
14. Damasio A (1994) "*Descartes' error: Emotion, reason, and the human brain*." New York: Gosset/Putnam.
15. Dolan RJ, Vuilleumier P (2003), "*Amygdala automaticity in emotional processing*." Ann N Y Acad Sci 985:348-355.
16. Dudai Y (2004) "*The neurobiology of consolidations, or, how stable is the engram*?" Annu Rev Psychol 55:51-86.
17. Eichenbaum H (2002) "*The cognitive neuroscience of memory*." New York: Oxford University Press.
18. Kosko, G., "*Fuzzy Cognitive Maps*," International Journal of Man-Machine Studies, 24: 65-75.
19. LeDoux JE (2002) Synaptic Self: How our brains become who we are. New York: Viking.
20. J. Rowley and R. Hartley, "*Organizing knowledge: An introduction to managing access to information*." Ashgate Pub Co, 2008.
21. L. Zadeh, "*A note on web intelligence, world knowledge and fuzzy logic*," Data & Knowledge Engineering, vol. 50, pp. 291-304, 2004.
22. S. S. Liao, J. W. He, and T. H. Tang, "*A framework for context information management*," Journal of Information Science, vol. 30, pp. 528-539, 2004.
23. M. Pinto, "*A grounded theory on abstracts quality: Weighting variables and attributes*," Scientometrics, vol. 69, pp. 213-226, 2006.

A Tool to Generate Computer Assisted Instruction Systems Through Hierarchical Classification

Richard Fox[1], Michaela Schleifer[1], Jennifer Cellio[2]
[1]Department of Computer Science
[2]Department of English
Northern Kentucky University, Highland Heights, KY 41099

Abstract - *Determining the cause of student learning errors is a form of diagnosis that permits teachers to tailor more effective lesson plans. Yet, teachers may become overwhelmed when there are hundreds of student learning errors to diagnose. Intelligent computer assisted instruction presents the teacher with a way to reduce the workload in performing the diagnostic task. However, constructing intelligent computer assisted instruction systems can be both time consuming and challenging. The Didactica tool provides an automated way to generate such systems. Each resulting system is specialized to a particular domain and problem. The effort to generate an intelligent computer assisted instruction system using Didactica is minimal, requiring just a hierarchy of learning errors, questions to test students, and patterns of expected and unexpected responses. This paper introduces the Didactica tool and demonstrates its use through examples in very different domains.*

Keywords: Intelligent Computer Aided Instruction, Artificial Intelligence, Tools, Diagnosis, Classification

1 Introduction

Determining the cause of student learning errors is a form of diagnosis. By diagnosing learning errors, a teacher can revise a lesson plan, introduce examples to correct the student, and create assignments specifically to adjust the student's learning behavior. However, the diagnostic process might be time consuming and when the teacher must correct a class full of student learning mistakes, the task may soon overwhelm the teacher. Intelligent computer-assisted instruction (ICAI) can help reduce the teacher's workload. ICAI (also sometimes referred to as intelligent tutoring systems, or ITS) is software that serves as an educational tool to instruct a student or test a student [1], and in the latter case, help diagnose student learning mistakes. ICAI systems range in complexity and capability from those that merely present knowledge and test students to those that can learn student behavior to those that can perform natural language discourse with the student.

Most ICAI systems today are model-based – that is, there is an underlying knowledge base of the domain knowledge and a model of the student's current state of understanding. Some form of Artificial Intelligence (AI) approach is then needed to present appropriate domain knowledge to the student and manipulate the student model based on responses obtained from the student. One popular approach is model tracing [2] where a series of production rules are used to directly test the student to see if the student understands the given concept. Rules are divided into varying levels of proficiency with some rules used to test for correct understanding and some to test for misconceptions. Rules can also be used to decompose a given problem into subproblems, testing the student at each step. Another common approach is through constraints [3] where the student model itself is based on constraints that must be true to ensure that the student has a proper understanding of the topic. In this case, the model is constructed during interaction with the student. If a constraint is violated, then the model must be updated to reflect the student's misconceptions. Rules are applied when a constraint is violated in order to further understand why the constraint was violated, and thus what the student is misunderstanding. Another, somewhat less utilized approach is that of Bayesian probabilities used to train the degree of uncertainty in the student model [4]. The model-based approach requires internally consistent knowledge and a sufficient amount of knowledge (breadth and depth) to handle both correct answers and student mistakes. While these approaches provide for a robust underlying system, they are expensive in terms of the amount of effort needed to construct and test the resulting ICAI systems.

While ICAI (or intelligent tutoring) systems have existed since the 1960s, a general-purpose ICAI system remains too complex to construct because of the depth and breadth of knowledge it would require, as well as capabilities such as natural language understanding, and learning. Therefore, ICAI systems are tailored to specific topics, such as math, computer programming, science and language. To alleviate the time consuming task of constructing an ICAI system, tools are available. Yet even with such tools like CTAT [5] and ASPIRE [6], construction time is often dozens to hundreds to thousands of development hours per hour of lesson plan [7, 8].

One way to reduce the effort required to construct ICAI systems is to limit their capability. If one wishes only to diagnose student learner errors, a classification-based approach can be applied (see for instance the intelligent tutor for diagnosing errors in understanding decimal math presented in [9]). Here, there is no deep underlying domain model, nor the need to explicitly model the student's knowledge, but instead a collection of learner errors, organized in a hierarchy. The use of classification for

diagnosis is commonplace in AI [10]. One particular form of classification is Hierarchical Classification [11], where one attempts to establish a concept as relevant by considering hypotheses of varying levels of specificity. The task is to search through a hierarchy of hypotheses in an attempt to find the most specific hypotheses that are deemed plausible. In a diagnostic situation, each hypothesis in the hierarchy represents a type of error (malfunction). When searching through the hierarchy of errors, if a hypothesis is established as plausible, then it is refined into more detail by attempting to establish any of the established node's children. The establish-refine process then is both efficient (only plausible hypotheses are refined) and provides the knowledge engineer with an easy approach to implementing an automated system.

This paper introduces the Didactica tool to automatically construct classification-based ICAI systems. The goal of this research is to provide a platform so that a teacher can insert and organize knowledge to generate an ICAI system. The generated ICAI system is then used to diagnose learning errors of students. To date, four ICAI systems have been generated using Didactica in the very disparate domains of dressage, computer programming, Spanish grammar, and identifying errors in writing patterns deemed too informal for traditional academic discourse. This paper is organized as follows. In section 2, the idea behind hierarchical classification is explained. In section 3, the Didactica tool is described. Section 4 describes Didactica's interface for constructing ICAI systems. Section 5 contains partial examples of two systems generated from Didactica. Section 6 offers conclusions and future work.

2 Hierarchical Classification

Diagnosis has often been considered to be a form of classification. In [12], Clancey defines the process that the MYCIN medical diagnostic system performs as Heuristic Classification, which comprises three separate tasks: data abstraction, data matching, and refinement. In [11], Chandrasekaran introduces a variation of Heuristic Classification called Hierarchical Classification. By separating out the classification component (refinement) from the overall process, Hierarchical Classification can be considered a more distinct element of diagnosis, and thus a building block in a problem solving system.

Hierarchical Classification requires a classification hierarchy. This is a taxonomy of concepts within the domain being classified. These concepts are organized from most general at the top of the hierarchy to most specific in the leaf nodes. The taxonomy might be of abstract concepts (e.g., parts of speech in natural language), physical objects to be recognized (e.g., types of vehicles), or diagnostic categories (e.g., liver disease). In the case of diagnosing student learning errors, the hypotheses are the types of misconceptions that a student might have when learning the topic. The hierarchy is organized with general diagnostic conclusions near the top of the hierarchy and specific errors in the leaf nodes. Figure 1 illustrates the types of errors that a student of dressage might have when learning to properly ride and control a horse. For instance, if a teacher (or trainer) determines that the student has a misunderstanding of proper dressage, this might be a problem with the orientation of the head of the horse, or the leg motion of the horse. On refining the diagnostic hypothesis to be a leg movement problem, one can further refine the misunderstanding as one of general movement or alignment. Eventually, one might identify the specific problem that the student has as being a failure to correct the horse's weight distribution.

In Hierarchical Classification, there is no commitment made to how a node (concept, object, hypothesis) is established as plausible. This is left open because there can

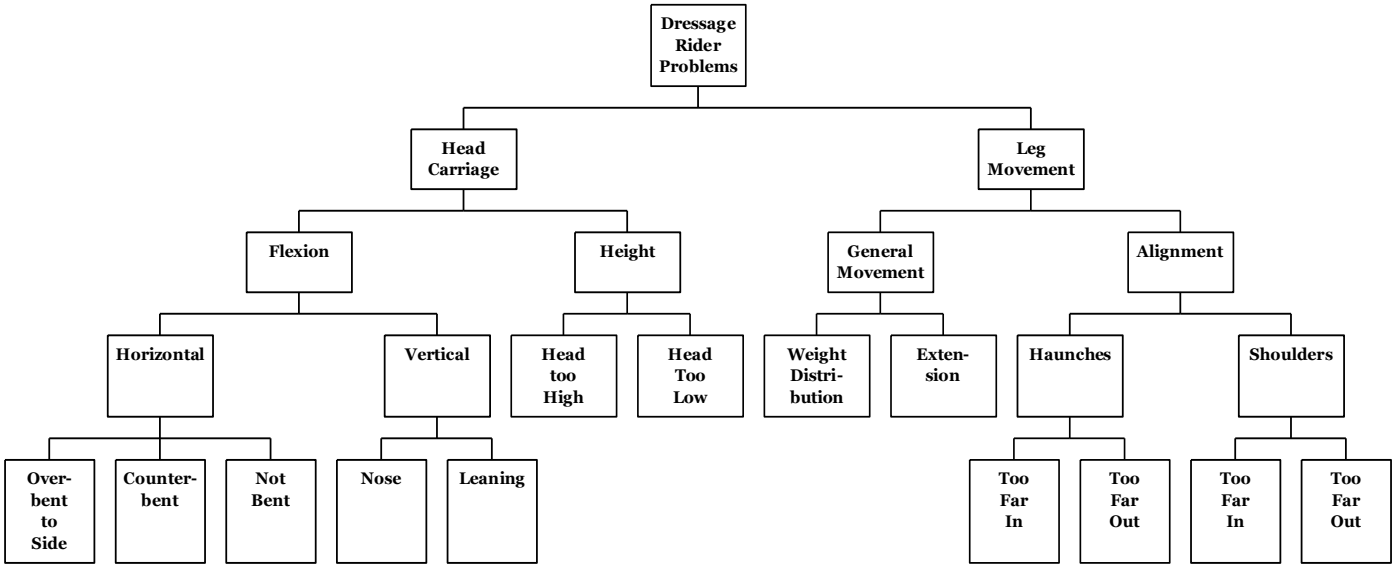


Figure 1: Mistakes of an Introductory Dressage Rider

be many possible approaches depending on the domain and the problem being solved. In general, one must obtain relevant input or data about the current situation (e.g., symptoms and test results in a diagnostic setting, an acoustic signal in speech recognition, student question responses in ICAI). The input is then used to generate a likelihood that the given node is found to be relevant. There are many different tools for generating a likelihood that data match a given concept in artificial intelligence, including Bayesian probabilities, fuzzy logic rules, rules with certainty factors, feed-forward/back-propagation neural networks (or simple perceptron), hidden Markov models and model-based approaches. Based on the result of the test for establishing the node, the node is either established as plausible, in which case the next step is to refine the node by recursively considering the node's children, or rejected and the children nodes are discarded from consideration.

To perform Hierarchical Classification, one needs the classification hierarchy and an algorithm to search the hierarchy. An "establish-refine" algorithm is suitable for most forms of diagnosis. In this algorithm, a node is considered for relevance by applying some test that can provide a likelihood, or plausibility, for the hypothesis. If the node is found relevant, it is refined by attempting to establish the node's children. If the node is not established, it is rejected and the child nodes are removed from consideration. The diagnostic process then is to search through the hierarchy of errors and identify any (all) that are plausible. Additional diagnostic knowledge could come into play to go beyond this classification if necessary (as might be the case in medical diagnosis) [13].

In the case of a ICAI system that is diagnosing student learning errors, a straight-forward approach to establishing an error hypothesis is to ask the student a series of questions related to the concept, and use the student's responses to determine whether the given concept is being understood or misunderstood. As an example, if one is trying to establish that a student has the misunderstanding of how to properly write a while loop in introductory programming, one can present the student with example loops and ask the student which ones work correctly. More specific questions would be linked with more specific diagnostic hypotheses. This is further illustrated in section 5.

In some cases, a hypothesis can be established through a single question. In other cases, establishing a hypothesis might involve asking multiple questions and examining the collection of answers. In evaluating a collection of student answers, there might be several different plausibilities that can be generated. For instance, if the student is correct on the first and second questions and wrong on the third, this pattern might indicate a weak likelihood of misunderstanding, whereas if the student misses the second and third questions, this might indicate a strong likelihood. For a node to be established, it must surpass some threshold value, as determined by the teacher. In some cases, a question's response might be used to establish multiple hypotheses. For instance, in the dressage domain, asking the student how far a horse has stepped can be used in determining general movement problems, weight distribution problems, and extension problems. Therefore, to simplify the time taken by the student in using the ICAI system, questions and answers are cached.

3 The Didactica Tool

The problem, as stated here, is that ICAI can be used to alleviate the workload of teachers by automating the process of determining where students have learning errors in a given topic. Building ICAI systems, though, can be time consuming and costly, even with the use of authoring tools. By limiting the problem to one of diagnosis through classification, a tool to automatically generate ICAI diagnostic systems can allow teachers to easily construct their own ICAI without a great investment in time or effort. Similar tools have long been available in AI to solve diagnostic problems [11]. The Didactica tool provides an interface for a teacher to generate the ICAI system. As the approach taken is based on Hierarchical Classification, the Didactica tool solicits three types of information from the user (the teacher): a classification hierarchy of learning errors, a series of questions that the teacher would use to test a student, and the pattern of responses that would indicate to the teacher that the student is suffering from each of the particular learning errors in the hierarchy.

As with a KADS tool [12], Didactica gives the user the ability to test out and manipulate the knowledge base as it is being constructed. At first, the user is presented with an empty hierarchy. The user specifies some of the error nodes and how they relate to each other (e.g., a parent-child relationship or a sibling relationship). As the user builds the hierarchy, the user is able to alter the relationships by adding new nodes, eliminating nodes and moving nodes as needed.

The user may at any time enter the knowledge needed to establish a given node. This knowledge consists of two types: test questions and expected patterns of responses. Some questions may be useful for multiple nodes in the hierarchy. In many cases, a node will require multiple questions. The user simply links a question to the node(s) that the question is intended for. A cache is used to store student answers in the case that a question is duplicated within the hierarchy. That cache is automatically consulted whenever any node is being established. If the cache does not contain the question, then the student is presented with the question, otherwise the answer is retrieved from the cache.

The user (teacher) is able to specify patterns of responses to the questions where each pattern can result in a different likelihood. As an example, consider a particular node that has three yes/no questions. The correct answers are yes, yes, no respectively. The teacher believes that if the answer to question 1 is incorrect then the user has at least some degree of confusion, but if the answers to question 1 and either 2 or 3 are wrong, then the student clearly misunderstands the

concept. In this scenario, there can be at least four patterns that the user specifies:

no no ? → very likely
no ? no → very likely
no ? ? → somewhat likely
yes ? ? → unlikely

The "?" in the pattern means "don't care", that is, it does not matter how the student answers. Expected answers in a pattern can be specified exactly, as in the case of yes/no or true/false questions, or with a specific value (e.g., how many times will this loop execute?), or as relational values such as "> 5" or "contains X".

The user is able to experiment with the knowledge in the system by running through various testing scenarios. There may be situations where, after knowledge is added, earlier knowledge is considered inconsistent. So the user is able to go back and edit the hierarchy, the questions, the expected answers and the patterns of responses as much as needed. At some point, when the user feels that the system is complete, then Didactica generates the ICAI system, a run-time executable. This resulting system is run by the students which, based on the student's responses, diagnoses the learning errors that the student has shown within the domain.

4 Didactica User Interface

To generate an ICAI system with Didactica, four primary menu options are utilized. First, the hierarchy of nodes must be created. Didactica allows the user to add or delete nodes, and to organize the hierarchy by defining parent/child relationships. The hierarchy is displayed by a simple list, with relationships denoted by indentation of the child node names. For example, the following is an excerpt from the dressage domain (refer back to figure 1):

Dressage Rider Problems
    Leg Movement
        General Movement
            Extension
            Weight Distribution
    Head Carriage

In addition to specifying the hierarchy, the user (teacher) must enter the specific questions that are used to test the student against each of the learning errors in the hierarchy. Didactica permits the user to display, add, delete, and modify questions at any time. Each question may be assigned a user-friendly name and category, which is intended to simplify the management of the question pool. This allows the user to review questions in a particular category and to reference questions by their names, instead of viewing their full text. The question menu also has an option to enable or disable case sensitivity when matching student responses to the questions.

Once both the hierarchy of nodes has been created and the questions have been entered, a linking function is used to associate the questions with particular nodes. The linking menu allows the user to specify conditions that represent the likelihood of a particular node being applicable. The conditions can be complex, with 'and' and 'or' statements used to combine specific questions, and nested parenthesized conditions permitted. For example, consider the following conditional statement:

(Q10!='4' and (Q12>'4' or (Q3>'2' and Q12='3')))

The statement utilizes each question's unique identifier followed by a comparator, and then a specified value. For this particular statement to evaluate to 'true', indicating that its associated node is a confirmed problem, the student's answer to question Q10 must be not equal to the value '4', and the remainder of the conditional must also evaluate to true. Didactica carefully manages the format of these statements to protect the user from accidental formatting errors.

The final step in creating a tutorial system with Didactica is to generate an executable tutorial that uses the node hierarchy, questions, and likelihood specifications. The user has the ability to configure the diagnostic output of the system to either verbose or succinct mode, and the system creates an executable that can be distributed directly to students.

5 Examples

This section presents two example ICAI systems generated from Didactica. The first system demonstrates a portion of the learner errors that an introductory programming student might have when learning about loop control structures. The second system demonstrates one type of error that an introductory college-level composition student might have when learning to write in higher education, that of being too informal. The dressage example presented earlier in the paper (see Figure 1) is a third system generated with Didactica. A fourth system has also been created to test Spanish grammar mistakes.

The loop error hierarchy is shown in Figure 2 (on the next page). This is a portion of a larger hierarchy that would cover errors in such concepts as method calling, variable usage and arrays. In this case, the loop misconceptions for an introductory student might be syntax errors (for instance, not understanding that a loop body in Java requires { } around it if the body consists of more than one statement, or that ; are used to separate the components in a for loop), errors with improper usage of loop variables, errors of improperly placed statements within, before or after a loop body, and errors dealing with nested loops.

After creating the error hierarchy, the user fills in questions, which are arranged to test each of the error hypotheses. Sample questions are shown in Figure 3. Some of these provide sample loops and ask such questions as the number of times the loop executes, or what the value of a variable is after the loop terminates. Other questions might be more general in nature, for instance asking what would happen if a loop variable is not initialized, or what the significance is of an infinite loop. Along with each question, the user must specify which specific node(s) will use the question. In Figure 3, for instance, the first question is used

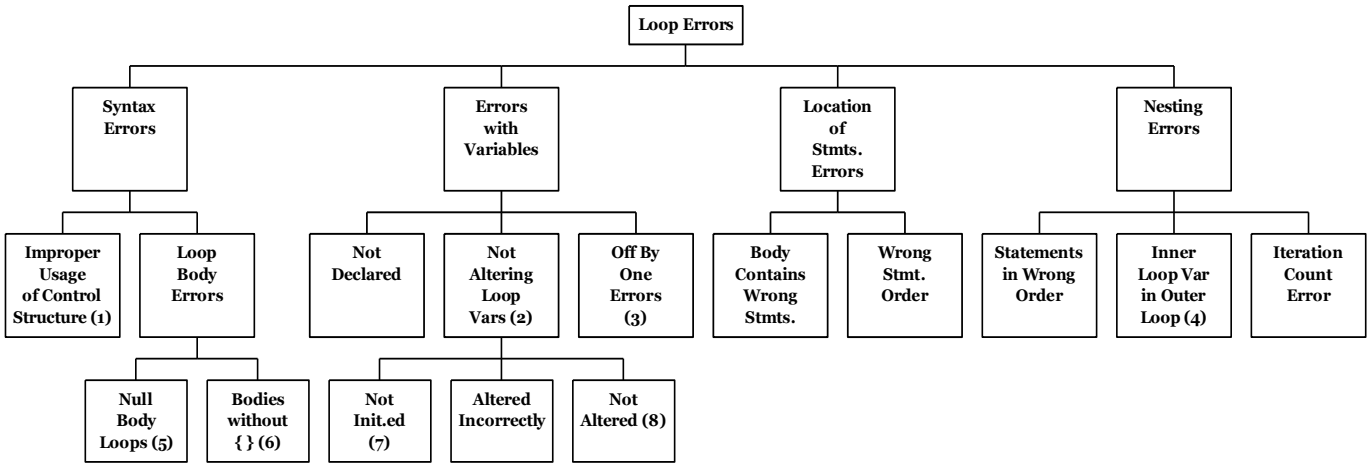


Figure 2: Loop Error Hierarchy (the numbers attached to some of the nodes are referenced in Figure 3)

by three different nodes (Improper Usage of Control Structure, Null Body Loops and Bodies without { }). As shown, some questions are used to establish or reject multiple error hypotheses. Finally, for each node, the user specifies patterns of expected responses that will determine whether that node is established or rejected.

Diagnostic output from Didactica for the Loop Hierarchy ICAI system, in verbose mode, can appear as follows:

Loop Error problems...
- Evaluating Errors relating to syntax errors...
- Ruling out all errors beneath "Misunderstanding Syntax".
  - Skipping:"Incorrectly Identify Loop Body".
  - Skipping:"Does not understand control structures".
- Evaluating Errors relating to variables problems...
- Ruling out all errors beneath "Errors altering loop control variable".
  - Skipping:"Failure to initialize loop control variable".
  - Skipping:"Errors altering loop variables ".
- Confirmed problem: "Off-by-one indexing errors". With hierarchical categorization of:[Common Loop Mistakes, Errors relating to variables, Off-by-one indexing errors]
- Evaluating Improperly located statements problems...
- Ruling out all errors beneath "Improperly located statements".
  - Skipping:"Including statements inside the loop that belong outside of the loop".
  - Skipping:"General misunderstanding regarding statement order".
- Evaluating Loop nesting errors problems...
  - Confirmed problem: "Terminating an outer loop prior to an inner loop". With hierarchical categorization of:[Common Loop Mistakes, Loop nesting errors, Terminating an outer loop prior to an inner loop]
  - Ruling out "Reusing the loop control variable of an outer loop".
  - Confirmed problem: "Loop iteration counting errors". With hierarchical categorization of:[Common Loop Mistakes, Loop nesting errors, Loop iteration counting errors]

Diagnosis complete.

Which line(s) comprise the body of this while loop? (1, 5, 6)

```
Line1:  int x=10;
Line2:  while (x>0) do ;
Line3:  System.out.println("The next value is:" +x);
Line4:  System.out.println("Done.");
```

1. Line2-3 2. Line 3 3. Line 4 4. None of the above

What happens if the loop control variable is not altered inside of the loop body? (1)

1. Compilation error 2. Runtime error
3. Loop never executes 4. Loop executes infinitely

Does the following While loop effectively count the first 10 items of List l? Yes or no? (2, 8)

```
int x=10;
while (x>0) {
    sum += list<x>
}
```

Assuming x is an integer initialized to 2. How many times will this loop execute? (3)

```
while (x > 1)
    x--;
```

What is wrong with the following code? (7)

```
int p, x=2;
while (p == x)
    p++;
```

1. Failure to initialize the loop control variable
2. Failure to alter the value of x within the loop
3. This code will produce an infinite loop
4. There is nothing wrong

True or False? It is a best practice to re-use loop control variables, especially when nesting loops? (4)

Figure 3: Sample Questions from the Loop Hierarchy (the numbers following each question reference nodes from Figure 2)

A second ICAI system is presented here to demonstrate the general-purpose nature of this approach. In this case, the domain is of introductory composition students' understanding of college-level writing. The domain is limited to a student's understanding of how to avoid using informal or conversational writing when it is not appropriate. Figure 4 (at the bottom of the page) demonstrates a portion of the error hierarchy and Figure 5 (to the right) provides a few of the questions that would be used in some of the hypothesis nodes.

At any time during the ICAI generation process, the user can revisit the hierarchy of nodes, the set of questions, and the conditional statements that specify the likelihood of a particular error. It is also possible to save a snapshot of the current system, or reload a previously-generated system for the purpose of performing modifications.

6 Conclusions

A teacher will diagnose student learning mistakes in order to redirect a lesson plan or create assignments that help the student correct those mistakes. ICAI is a very valuable way to help automate the process so that the teacher is not overwhelmed by having to test and correct a large number of students. But ICAI systems can be time-consuming and therefore expensive to construct. The Didactica tool provides a platform to automatically generate ICAI systems. The systems generated use the AI strategy of Hierarchical Classification to perform the diagnosis. The Didactica tool has been used to generate ICAI systems in four very different domains: understanding loops by introductory programming students, introductory student mistakes in dressage, introductory composition student's understanding of college-level writing, and most recently, Spanish grammar. The dressage domain might be thought of as unusual because the errors are not so much a student's misunderstanding of topics, as it is a student's ability to properly control and guide a horse.

[Sample text provided, each sentence is numbered. Some sentences are informal, for instance by providing run-on sentences, or by using informal language such as beginning sentences with "Now" or "I mean".]

In which line(s), does the text contain conversational cues?
1. None of them
2. Sentence 3
3. Sentences 3 and 4
4. All of them

Are there any sentences which should be divided into multiple sentences?

Which of the following is true about sentence 2?
1. There should be a comma between the word "barn" and "and"
2. There should be a new sentence started after the word "barn"
3. The word "and" should be omitted and a new sentence started at that point
4. There is nothing wrong with the sentence

True or false? In general, sentences should never have more than 1 clause.

True or false? The use of the word "you" is acceptable in essays but not in reports.

Figure 5: Some Informal Writing Error Hierarchy Questions

The idea is that through the Didactica tool, a teacher has a far easier way to generate ICAI systems that can be tailored for the specific topic being covered in class. As shown here, the construction of the ICAI systems is a matter of organizing a classification hierarchy and providing test questions and responses. To demonstrate the usefulness of the approach, the loop system was generated in just three hours by a graduate student in computer science. Work continues on the Didactica tool in several ways. First, a graphical user interface is being constructed to ease the specification of the hierarchy

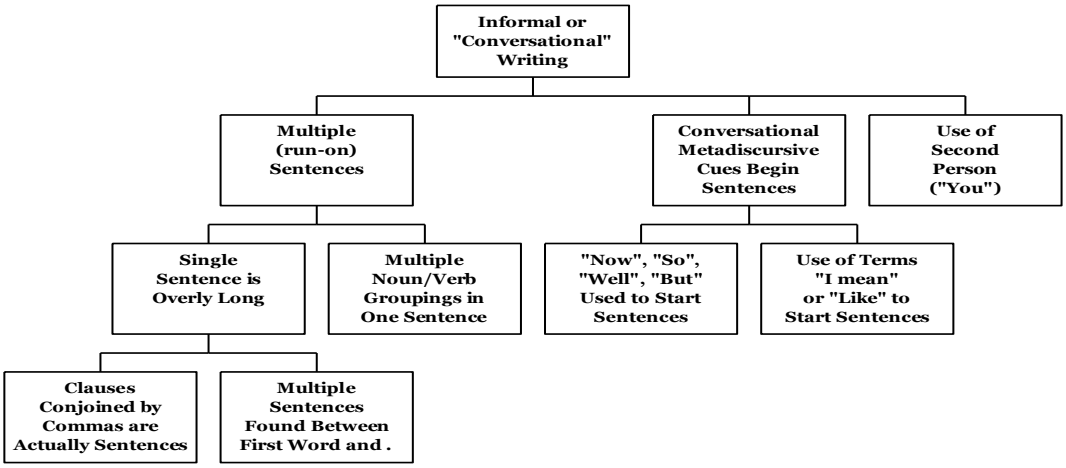


Figure 4: Errors Hierarchy of Informal Writing Techniques in College-Level Introductory Writing

(currently, it is text-based). Second, the resulting diagnosis is being expanded to provide more useful content to the teacher (e.g., what questions were wrong that led to the given diagnosis). Third, the composition ICAI system will be tested by undergraduate students in an English composition course as further demonstration of the utility of this approach. It is possible that the Loop diagnostic system will be tested in future introductory programming courses. Fourth, additional ICAI systems are being generated. At the time of this writing, a fourth system, to diagnose grammatical mistakes in Spanish, has been completed

7 References

[1] Rickel, J.W., Intelligent computer-aided instruction: a survey organized around system components, in *Systems, Man and Cybernetics, IEEE Transactions on* , vol.19, no.1, pp.40-57, Jan/Feb 1989 doi: 10.1109/21.24530

[2] Anderson, J.R., & Pelletier, R. A Development System for Model-Tracing Tutors. In L. Birnbaum (Ed.) Proceedings of the 1991 International Conference on the Learning Sciences (pp. 1-8). Charlottesville, VA: AACE, 1991.

[3] Mitrovic, A., Koedinger, K., M., & Martin, B. A Comparative Analysis of Cognitive Tutoring and Constraint-Based Modeling. In P. Brusilovsky et al. (Eds.) Proceedings of the 9th International Conference on User Modeling (UM2003) (pp. 313–322). LNAI 2702. Berlin Heidelberg New York: Springer-Verlag, 2003.

[4] Butz, C., Hua, S. and Maguire, R. A web-based bayesian intelligent tutoring system for computer programming. Web Intelligence. and Agent Systems. 4, 1, 77-97, 2006.

[5] Aleven, V., McLaren, B. M., Sewall, J., & Koedinger, K.. The Cognitive Tutor Authoring Tools (CTAT): Preliminary evaluation of efficiency gains. In M. Ikeda, K. D. Ashley, & T. W. Chan (Eds.), *Proceedings of the 8th International Conference on Intelligent Tutoring Systems* (ITS 2006), (pp. 61-70). Berlin: Springer Verlag, 2006.

[6] Mitrovic, A., Martin, B., Suraweera, P., Martin, Zakharov, K., Milik, N., Holland, J., and McGuigan, N. ASPIRE: An Authoring System and Deployment Environment for Constraint-Based Tutors, the *International Journal of Artificial Intelligence in Education*, Vol 19, Number 2, pages 155-188, IOS Press, 2009.

[7] Koedinger, K., Aleven, V., Heffernan, N., McLaren, B., and Hockenberry, M. Opening the Door to Non-programmers: Authoring Intelligent Tutor Behavior by Demonstration, in Intelligent Tutoring Systems, Volume 3220, p. 7-10, Lester, J., Vicari, R., and Paraguaçu, F. editors, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2004.

[8] Murray, T. Authoring intelligent tutoring systems: An analysis of the state of the art, in the International Journal of Artificial Intelligence, 10, p 98-129, 1999.

[9] Isotani, S., McLaren, B., and Altman, M. Towards Intelligent Tutoring with Erroneous Examples: A Taxonomy of Decimal Misconceptions, in Intelligent Tutoring Systems, Volume 6095, Aleven, V., Kay, J. and Mostow, J. editors, p. 346-348. Lecture Notes in Computer Science, Springer Berlin/Heidelberg, 2010.

[10] Fox, R. and Davidson, E. Classifying User Behavior in Linux Sessions, in *The Proceedings of the 2008 20th IEEE International Conference on Tools with Artificial Intelligence*, Volume 2, p. 202-205, IEEE Press, 2008.

[11] Chandrasekaran, B. Generic tasks in knowledge-based reasoning: High-level building blocks for expert system design, in IEEE Expert, p. 23-30, Fall 1986.

[12] Clancey, J., Heuristic Classification. Artificial Intelligence, 27:289-350, 1985.

[13] Josephson, J., Josephson, S. Abductive Inference: Computation, Philosophy, Technology, New York: Cambridge University Press, 1994.

[14] Waldner, Jean-Baptiste. Principles of Computer-Integrated Manufacturing. John Wiley & Sons, September 1992.

Relational Modeling in Social Media

David Alfred Ostrowski
System Analytics
Research and Innovation Center
Ford Motor Company
dostrows@ford.com

Abstract

Relational Modeling has been gaining traction towards the support of collective intelligence. Here, we explore the performance of Relational Classification as compared to both supervised and unsupervised Machine Learning methodologies. These techniques are applied to Social Media-based data sources with the goal of understanding consumer behavior. We demonstrate that Relational Modeling can provide comparative performance to Machine Learning methods and demonstrate even higher accuracy when an expanded social network is leveraged towards the learning process.

Keywords – Relational Modeling, Social Media, Machine Learning

1. Introduction

Web based data has provided substantial opportunities in the area of collective behavior, lending support to the development of Relational Modeling. [1][2][3] In particular, Social Media has become of special interest in the area of determining consumer behavior. [4][5] Considered first at an individual level, the information gathered can support the identification of trends as well as related content.

At a higher level, social network associations can present even more intuition around the identification of the behavior of individuals or groups. Such associations may be indicated through a number of means including friends or followers as well as higher associations such as identified values shared between them. In consideration of associated groups, trends can not only be identified but strengthened. This can provide support to market segmentation and collective behavior prediction.

As considered in the context of linked data sources, Relational modeling can be regarded as a very powerful and relatively unsophisticated methodology. Opportunities in this area are widespread as individual associations spread through a network can lend support to each other. Within given associations, many dimensions of relationships may easily exist. Such links may also be explored beyond many different data sources. Considered in the context of social networks, such relationships can exist within a specific site or be linked between several different sources of data.

In this paper, we detail an application of Relational Modeling as it is applied to the classification of individuals in Social Media. We consider a variant of Relational Modeling termed as Relational Neighbor (RN) which is a simple relational model that predicts only on class labels of relational neighbors using no learning of inherent attributes. [3] We consider our classification towards samples of accounts in Facebook. We show that it performs well as compared to more complex models such as more traditional forms of machine learning algorithms including Bayesian Learning and Clustering.

A number of goals exist within our research. First, we consider a test framework of existing classification methods towards evaluation of Relational Modeling as applied to Social Media. We also consider our effort towards overcoming noted deficiencies in the area of Social Media-based data sources which include incomplete as well as improperly distributed data sources. [6] This work also serves to motivate the inherent flexibility of the Relational Learning approach supporting the creation of relational instances to be considered among different experiment designs. Among such designs, we consider link dependencies as the key factor supporting our directions towards future work in this area especially in the context of dynamic data sources.

The remainder of this paper is as follows: In Section Two we detail the current work in the field. Section Three provides a detailed examination of the

application of our methodology. Section Four presents two separate case studies and Section Five presents our conclusion.

2. Current Research

In basic analysis of the Relational Model (Provost , Perlich and Macskassy) examined the construction of the relational approach as it is considered among associated background knowledge. As such, they investigated the potential for dimensions to be explored or considered towards the context of a proposed model. These explorations were set as the premise towards the development of a common framework to characterize relational learning which included consideration of relational neighbor- based models as well as evaluation of probalistic modeling. [2]

By studying collaborative relationships, (Haung) demonstrated that entities tend to be alternatively consistent over time. By investigating and applying such properties, he was able to propose two mixed group discovery algorithms as well as generic models which incorporate group linkages in the context of social network analysis to support relational learning.[6]

A number of efforts in Relational Modelling have helped to resolve abnormalities in data. Addressing issues of imbalanced data (Ghanem, Venkatesh and West) proposed a probalistic approach towards the inbalanced class problem. By presenting a new model PRMs-IM, they were able to identify at-risk students within a real university database. Through this test case, they were able to demonstrate a means to overcome situations where one class of data was over-represented. [7]

Addressing issues of incompletely labeled data, Lin and Cohen were able to apply a weighted-vote relational classifier to semi-supervised learning in the context of a weighted graph. Here, they demonstrated optimal classification accuracy by establishing authoritative instances as training seeds while applying a reduced amount of training data. [8]

Our investigations into Relational Modeling also references the related field of link mining strongly supporting activities of predicting participation of events and expected consumer behavior.[9] Here, (Macskassy) relied on an inference-based model which was termed as 'guilt-by-association' in which numerical data was classified in a transductive setting. Here, by associativity, web based data (through links) was leveraged to support the classification process. [10]

Initiated from earlier efforts concentrated in the category of Web Mining, a number of methodologies have influenced Relational Learning frameworks (towards classification) including PageRank and HITS. [11][12] Building on these methodologies, Bharat, and Henzinger and Chakraborti proposed variations of these techniques that exploit web page context to weight pages and links based on relevance.[13][14]

Relational learning has also been strongly associated with Semantic Web-based technologies. (Rettingan, Nizeler, Tresp) applied statistical relational learning in conjunction with ontologies specified by formal logic. Here, they employed hard constraints to machine learning thus supporting a predictive performance within their classification framework. [15]

(Thushar and Thilagam) also relied on Semantic Web technology for the identification of associations within a social network. Among the heuristics that they employed towards this development included the unpredictability of an association due to the participant of a property in multiple domains. They also examined considerations of a match between user specified keywords and popularity of nodes involved in a sequence to rank association. [16]

(Gloor, Nann, Schoder) developed a sophisticated semantic social network analysis tool called Condor which incorporated a number of methodologies including social network analysis and information filtering methods determining the betweeness centrality of actors and weighing the context by the social networking position of actors. Through this application, they were able to extract and predict long term trends on the popularity of relevant concepts such as trends, movies and politicians. [17]

(Zhou, Chen,Yu) integrated an ontology-based social network along with a statistical learning method towards Semantic Web Data. This involved utilizing an extended FOAF (Friend-of-a-friend) ontology applied as a mediation schema to integrate social networks and a hybrid entity reconciliation method to resolve entities of different data sources. [18]

Methodologies such as clustering have been demonstrated to be of high utility when applied towards Relational Learning problems. Towards this means, Kirsten and Wrobel proposed a methodology that presented a bottom-up agglomerative algorithm for first order representation that relies on distance information towards conceptual clustering. Using their technique, they were able to consider the accuracy of dedicated predictive learners. [19]

Social dimensions have also been considered in terms of understanding the learning process in the context of Social Media. Towards this means Tang and Liu proposed an edge-clustering scheme to extract spare social dimensions. Here, they worked

through the hypothesis that connections of the same network are not homogeneous and relied upon the determination of Social dimensions to address such heterogeneity. [17]

3. Methodology

Relational Modeling is predicated on the estimation of class probabilities. Our methodology is based on the assumption that relations exhibit homophily (where related entities are believed to have similar qualities). We also consider that entities' class labels are identified with same linked structure.

The Relational Neighbor classifier estimates the class membership probability indicated by P (c | e) as the entity c belonging to a class e. A weight factor is assigned between the given entities e and e_i. The variable Z is assigned to the summation of the entities with the approximation of a given label. Here, D_e exists as the set of entities linked to e. Under the circumstances that D_e is unknown, the missing class labels that RN will estimate is based on the class of prior (known) labels..

$$P(c|e) = \frac{1}{Z} \frac{\sum}{\{e_{j\in} \in D_e \mid label(e_j) = c\}} w(e, e_j)$$

Figure 1. Class Membership Probabilities.

Towards support of unknown (neighbors) class labels – the RN algorithm is applied in a recursive fashion. As such, the RN classifier would be re-applied at a successive layer towards the resolution of a missing label. Here, RN^{i+1} would be considered as the model at iteration *i* where RN^i is the current node which maintains a direct link to the next iterative level. At each degree of separation, linked nodes may be applied to successive associations at each level to backfill the appropriate missing nodes. We consider our weight factor along a normal distribution towards the identification of missing node labels as we consider levels of separation from our initial nodes in the network.

3.1 Application

In our framework, we examined sampled entities (social networking ids) with no explicit affiliations with each other. All entities were selected with known class labels. Here, the Relational Model considered entities that are directly associated to each other through hard links as determined by friend associations. (figure 2)

From the first-level associations between the entities we apply the RN algorithm to friends of friends (second to 'n' level association) to support both the means of identifying the associated class labels as well as strengthening associated class label distinction. From the support of identification of missing labels, the friend relationships are explored until a meaningful sample of data (ie friend links along with associated labels) from which a distinction may be made

Our premise is that networks can maintain a higher level of predictability in terms of supporting unknown class information. By proposing associations through originally missing nodes, unbalanced or incomplete information can be interpreted in a more accurate fashion. We also consider this towards the evaluation of entities which may be missing as a reinforcement effect through stronger associations confirming or strengthening existing features. Also, we consider the support of multiple relationships and values to be reinforced by the initial support of a group of friends.

The adjustment of weights are considered in our framework within the n-degree relationships. While a diminishing effect is explored along a normal distribution, we consider extra factors such as normalized percentage of associated neighbor entities at each n+1 degree relationship. Towards the evaluation of our data results we employed the standard means of recall as measured between our algorithms.

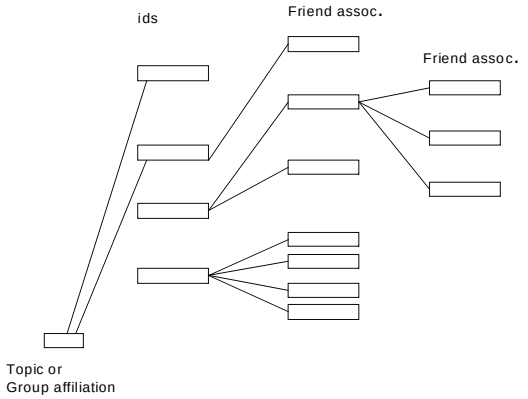


Figure 2. Social Network Association.

4. Implementation

4.1 Test Case I

We consider an application of the Relational Neighbor classifier as applied among a set of 2014 publically available Facebook accounts. In this case, we apply our classification efforts towards interest groups. Within each topic, a taxonomy was derived from which to make word associations to a feature category. Starting with identifiers to existing labels, the accuracy of our original entity set was measured against the result of the relational classification process. As noted in figure 3, the accuracy was grouped along the x axis by the normalized percentage of missing labels among the friend associations.

Our approach is compared to a Bayesian classifier considering our taxonomy-based feature sets applied in a 10-fold validation. We also compared them to a k-means clustering based algorithm. Here, the k-value was established to the appropriate feature set size (6) with the largest set predicting the feature of the original (parent) entity.

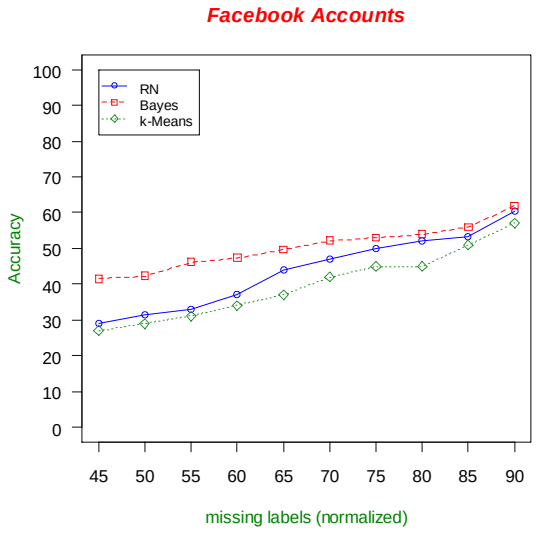


Figure 3. Accuracy comparison between RN, Bayes and k-Means Clustering

The RN classifier was evaluated between ten categories (45-90%) of the missing labels with the highest score of 62% among 90% labeled classes. The Bayesian model performed best on the lower (measured) missing labels among neighbors surpassing the RN by only one percentage point (at the lowest level of missing labels). The k-means clustering performed lowest within a slim margin increasing over the percentage of identified labels. The Bayesian model's performance was attributed to knowledge obtained from the training against the known 'friend' relationships. This resulted in a much higher performance among the higher amounts of missing labels.

4.2 Test Case II

The second case considered the same initial data sample along with additional associated 'friend' entities (derived from the local social network) for estimation of the missing labels. At the point of each missing label, the RN classifier is applied to the associated entities via successive friend links to derive a solution.

Our results are comparatively applied to the unchanged Bayesian and clustering metrics. At this point, the RN classifier made substantial gains in performance in particular over the categories of higher missing labels surpassing the accuracy of the Bayesian classifier. The results demonstrate that the knowledge obtained from the relational associations allows for a similar performance in classification to the learned relationships from the supervised Bayesian classification.

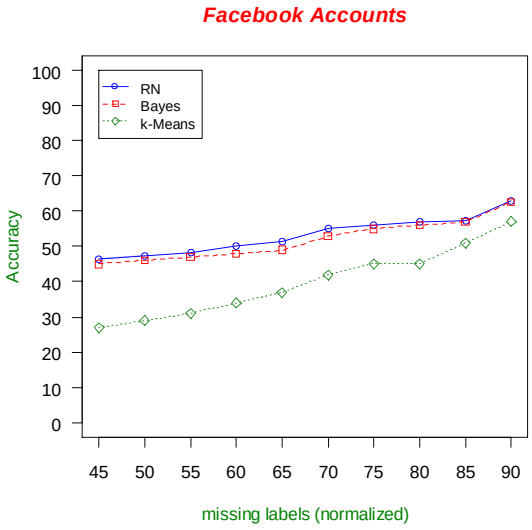


Figure 4. Accuracy comparison between RN (utilizing social network), Bayes and k-Means Clustering

5. Conclusion

An RN classifier has been presented for the means of determining the preferences/ interests of an individual among the taxonomy considered. Our results were able to detail a high level of accuracy our friend relationships supported a high level of topical information regarding interest categories. This

performance was measured to comparative machine learning based approaches as well as distance based metrics from a cluster based algorithm. We were able to demonstrate very close performance (within 1 percentage accuracy at the high end of consideration of missing nodes) . On a second approach, the network analysis was compared among identifiers that have a common topic association as well as to friend associations between the means of having friends of friends that are compared utilizing the iterative graphical based search. This approach surpassed the machine based methodologies providing a high level of return.

Towards our future work we consider the incorporation of Bayesian and Relational modeling towards the most effective means of classification. Future steps also include the support of Relational modeling towards higher level feature derivation as well as confirmation.

6. References

[1] Sofus A. Macskassy and Foster Provost, A Simple Relational Classifier,Proceedings of the Second Workshop on Multi-Relational Data Mining (MRDM-2003) at KDD-2003, pp 64-76

[2]F.J. Provost, C. Perlish, and S.A. Mackassy, Relational Learning Problems and Simple Models, In Proceedings of the Relational Learning Workshop at IJCAI-2003, 2003.

[3]Popescul, A., L. Ungar, S. Lawrence, D. Pennock (2003). Statistical Relational Learning for Document Mining. In *Proceedings of IEEE International Conference on Data Mining* (ICDM 2003).

[4] P. Domingos and M. Richardson. Mining the network value of customers. In Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, , San Francisco, CA, 2001. ACM Press. pp 57-66

[5]M. Richardson and P. Domingos, "Mining Knowledge-Sharing Sites for Viral Marketing," Proc. 8th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining

[6]Huang, Shu, Mixed Group Discovery: Incorporating Group Linkage with Alternatively Consistent Social network Analysis, 2010 IEEE Fourth International Conference on Semantic Computing, September 2010, pp. 369,376

[7]Ghanem, A.S.; Venkatesh, S.; West, G.; Dept. of Comput., Curtin Univ. of Technol., Perth, WA, Australia Pattern Recognition, 2008. ICPR 2008. 19th International Conference on Dec. 2008 pp 1 - 4

[8] Lin, F.; Cohen, W.W.; Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on 9-11 Aug. 2010 pp 192 – 199

[9] Getoor, L. and C. Diehl (2005). Link Mining: A Survey. *SIGKDD Explorations*, December, 2005, Volume 7 Issue 2

[10] Sofus A. Macskassy, Improving within-network classification with local attributes, In Workshop Papers on Text-Mining and Link Analysis (Textlink) at the 20 th Int'l Joint Conf. on Artificial Intelligence IJCAI, 2009

[11]L. Page, S. Brin, R. Motwani, T. Winograd, The PageRank citation ranking: Bringing order to the Web. ", Technical Report, Stanford Univesity, 1998

[12]J. Kleinburg, Authoritative sources in a hyperlinked environment. Journal of the ACM 46(5): 604-632 1999

[13]K. Bharat , M.R. Henzinger, Improved algorithms for topic distillation in a hyperlinked environment. In ACM SIGIR International Conference on Research and Development in Information Retrieval, pp 104-111, 1998

[14]S. Chakrabarti, B.Dom, and P.Indyk, Enhanced hypertext categorization using hypelinks. In SIGMOD International Conference on Management of Data pp 307-318, 1998

[15] Achim Rettinger Matthias Nickles, Volker Tresp Statistical Relational Learning with Formal Ontologies, ECML PKDD '09 Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part II

[16] Thushar, A.K.; Thilagam, An RDF Approach for Discovering the Relevant Semantic Associations in a Social Network P.S.; Advanced Computing and Communications, 2008. ADCOM 2008. 16th International Conference on *Pub: 2008 , pp 214 – 220*

[17] Gloor, Peter A, Krauss, Jonas S., Nann, Stefan, Fischbach, Kai and Schoder, Detlef, Web Science 2.0: Identifying Trends Through Semantic Social Network Analysis (November 11, 2008).

[18] Chunying Zhou; Huajun Chen; Tong Yu;
Tools with Artificial Intelligence, 2008. ICTAI '08. 20th IEEE International Conference on Volume: 1 Learning a Probabilistic Semantic Model from Heterogeneous Social Networks for Relationship Identification 2008 , pp 343 - 350

Towards an Automated Composer Of Popular Country Music

Jim Suruda, Norman Carver
Department of Computer Science, Southern Illinois University, Carbondale IL, USA

Abstract— *In the context of a software tool that can generate novel melodies in the popular country genre, this article describes the development of such a system. Algorithmic composition systems typically generate musical works and leave the evaluation of the output as a task for a human listener. This article describes an attempt to use machine learning techniques to evaluate the quality of the generated melodies. We describe the architecture of a prototype melody generation and evaluation system, which uses extracted features from MusicXML to judge the quality of machine generated melodies. Experimental results, including rankings of generated songs compared to currently charting country songs, are reported.*

Keywords: machine learning, composition, music, artificial neural networks, genetic algorithms

1. Introduction

The first computer program ever written might be one that calculates Bernoulli numbers, written by Ada Lovelace for Charles Babbage's computing engine. In her notes on the computing engine, Lovelace speculated about what further problems a more powerful computing engine might solve:

> "[computers] ... might compose elaborate and scientific pieces of music of any degree of complexity or extent" [4]

So, interest in computer music composition music begins with the first computer science research paper ever published in English. Researchers continue to explore new algorithmic composition techniques every year, from jazz solo construction to real-time accompaniment systems to AI applications that generate concertos. But researchers rarely conduct machine evaluation of generated music. A recent and exhaustive book on the subject of algorithmic composition makes only a brief one-paragraph mention of the evaluation of generated material, and cites no research in evaluation at all. [10] Despite the lack of mention by Nierhaus, there has been some work on the evaluation of generated work, [8] [13], and many more avenues of exploration still beckon.

Music generation and evaluation is a naturally appealing subject for research because even a prototype system can produce novel and interesting compositions, giving an exciting glimpse at the future of AI, when intelligent systems might begin to emulate and expand beyond human's creative abilities. More practically, efforts that uncover generation rules or aid in feature extraction from the highly structured but well-defined data of music could have applications in related fields. The areas of medical data, imaging, and natural language all have dense and complex data sets in which important features are not always obvious.

We chose music generation as the domain for our testbed system because a working prototype of a generation and evaluation system makes an excellent springboard for further research in machine learning. Judgements about the quality or catchiness of music are naturally imprecise and fuzzy, and lend themselves well to soft computing and AI techniques. Working within an existing genre simplifies compositional complexity, and our geographic and cultural proximity to Nashville Tennessee made popular country music a compelling choice for genre.

This paper describes the initial implementation of ACME (The Automated Country Music Engine), a testbed we have developed to explore computer generation and evaluation of popular country music. The initial implementation has demonstrated the feasibility of using a simple stochastic context-free grammar to generate a variety of novel melodies within a genre, and the possibility of using extracted features rank the fitness of the generated works.

2. Related Work

Researchers had been developing algorithms for generating melodies and producing harmony accompaniment long before computers existed. One of the earliest examples of melody generation is the Musikalisches Wurfelspiel, published in book form in 1792 and attributed to Mozart. The book contained small sections of music that the user would select by rolling dice. Users would play the concatenated sections as a single, novel work of music. Algorithms for harmony accompaniment date from 1725, when Johann Fux published a system of rules for creating counterpoint accompaniment to existing melodies. [5]

Within years of the development of the first electronic computers, researchers were writing programs to generate music. One of the first to do this was Henry F. Olson, who in 1950 extracted first and second order Markov models of note transitions from the works of Stephen Foster and used them to generate novel melodies in the style of Foster. [11] Also in the 1950s, Noam Chomsky's work on generative grammars for natural language inspired music researchers to develop generative grammars for music. Lerdahl and Jackendoff built on Chomsky's work in the 1980s to develop an ambitious grammar designed to generate any type of tonal music [9] Since then, researchers have applied techniques from nearly

every area of AI and machine learning to the creation of music works: cellular automata, transition networks, genetic algorithms, and rule-based systems. But few have attempted to quantify the quality of the generated works.

Klinger and Rudolf [2006] used an artificial neural network to rank the fitness of computer generated melodies. [8] They extracted a small set of features from melodies and used those along with user-supplied evaluations to train a feed forward neural network to rate the melodies. They had difficulty creating a large enough body of training data because they relied on users to listen to and rank each training example, which was time consuming.

3. Problem Definition

When country music songwriters write songs, they often create and reject a number of melodies in the process of writing one song. Writers repeatedly create and reject series of notes until they find something "catchy" or interesting to incorporate into a song. [3] Of all the songs they write each month, writers only submit a few of the best to their publishers. [7] Publishers select a few of the most promising songs to pitch to artists, who may choose some of the songs to record for an album. Of all the thousands of albums released each year, only a few songs get airplay on radio stations. So for each song that makes it to the radio, a huge number of songs are created, and only a few of the "fittest" songs are selected from the pool for recording and popular rotation. This is the process we modelled for our prototype music generation system: create a large population of songs, then use an evaluation system to select the best compositions. It is those songs that are undeniably excellent that are in demand, not those that are reasonably good. Tom T. Hall once said, on the subject of "pretty good" songs "... if we could make a career out of writing pretty good songs there'd be a lot more people in the business than there are." [7]

When a publisher chooses a song that they would like to present to a performer, they create what is called a "demo" of the song. The demo is a sample recoding of the song usually performed by studio musicians and a vocalist, created to showcase the song to a performer. The studio musicians play from a lead sheet, which contains only the melody notes, lyrics, and chord symbols. [3] During the demo session the studio musicians and producer create the song arrangement; the harmony, bass parts, backup vocals, and other instrumentation. Thus, the essential deliverable produced by the songwriter is the lead sheet: melody, lyrics, and chord symbols. Since our system does not attempt to create lyrics, the output will be melody notes and chord symbols. Figure 1 shows an example of a lead sheet for a popular country song.

MusicXML is a standard format for lead sheets and is supported by a variety of scorewriting and music editing tools, so we chose to have the project produce and evaluate songs in MusicXML format. The system should be able to generate a large number of unique and original songs and evaluate them

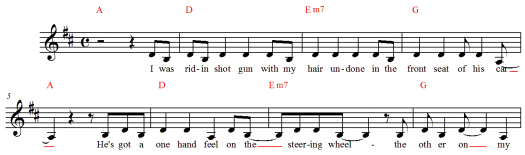


Fig. 1: Section of a Lead Sheet.

based on a heuristic that has some relation to the evaluation that occurs when human-composed songs compete for radio airtime. If our project is successful, the top-rated songs the system generates should be of equal quality to those composed by a human expert. This success is dependant on the ability of the generation system to produce successful songs and on the ability of the evaluative system to successfully recognize the best compositions. The acme of success for our prototype would be for an artist to record one of the computer-generated songs, and for the song to garner radio airtime.

4. Project Goals

We implemented our system, called the Automated Country Music Engine (ACME), to provide the following capabilities:

1) Generate melodies
2) Add chord accompaniment
3) Write MusicXML
4) Read existing MusicXML
5) Extract features from MusicXML
6) Train a neural network to rank songs
7) Evaluate generated songs

Our goal for this iteration of ACME was to create a complete, working system. Breaking the project up into subcomponents based on the capabilities listed above made it possible to attack the problem in independent sections. We chose to implement ACME in C#, because the ergonomic IDE facilitates rapid prototyping, because of its strong XML and string manipulation capabilities, and partly due to philosophical preference for a statically-typed language.

5. Implementation

Figure 2 illustrates the high-level architecture of ACME. In the following sections, we will briefly describe the implementation of each major capability.

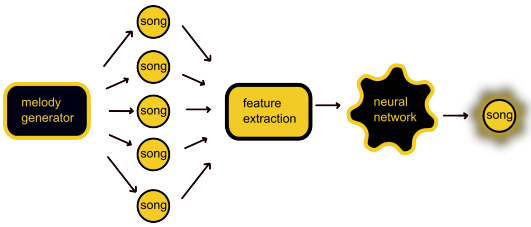


Fig. 2: ACME System Overview.

5.1 Generating Melodies

If our composition system is capable of successfully evaluating melodies, the melody generator does not need to be especially good on average, as long as it can generate a large variety of unique melodies within the popular country genre. The space of all possible musical compositions is obviously enormous, but the characteristics of typical country music compositions are relatively tightly constrained, so the subspace of reasonable country songs is much smaller. It is worth considering how large this subspace is to get a sense of the difficulty of the problem of generating popular country music.

A representative country song might consist of only three main sections (verse, chorus, and bridge) that are repeated. Each section could be about 8 bars of music, with 8 notes or rests possible per measure. Using a scale of eight notes plus a rest, there are 9 possible tones available. So, the search space of all possible country songs is roughly 9^{3*8*8} unique melodies. If a system could evaluate a million of the songs in the search space per second it would take $1.7 * 10^{55}$ years to process all possible songs. Enumeration of all possible works is not a practical generation strategy. Other, less constrained genres would have even larger search spaces, so popular country is still an attractive choice for an experimental domain.

For our initial implementation, we settled on a generative grammar to produce melodies since this made it easy to represent the types of rules found in songwriting books. Instead of enumeration, we settled on a simple generative grammar to produce melodies. Generative grammars were introduced by Noam Chomsky in the 1950s [2] as a way to describe the hierarchical structure of the human language. They can also be used to model the meta-structure of music, and to serve as a production guide for generating melodies. This hierarchical deconstruction of music was actually first proposed by Heinrich Schenker in the 1930s and was a basis for the hierarchical system for the construction of music later proposed by Lerdahl and Jackendoff. [9]

Although an implementation of Lerdahl and Jackendoff's grammar might produce a wide variety of novel melodies, they would certainly not be limited to popular country melodies. At the opposite end of the spectrum, popular country songwriting books often provide no algorithmic or generative rules for producing melodies. For instance, despite its title, Tom T. Hall's book "How I Write Songs", contains no description of methods for creating melodies. [7] Rather than devise our production rules *a priori*, we based ACME's generative grammar on those described by Stephen Citron in his book "Songwriting." [3] Although written for the human songwriter, Citron's melody generation rules sketch the outlines a stochastic context-free grammar that forms the core of ACME's generation system.

The terminal symbols in our musical context-free grammar are notes and rests. ACME limits itself to notes from a single major or minor scale for each composition. Although the conventions of popular country music allow accidentals and modulation, for simplicity ACME does not currently use notes outside the scale, and does not change the tonic of the within the melody. Assembling a series of notes from the lead scale into what we call a melodic cell is Citron's first production rule for melody construction.

To select a sequence of tones for the melodic cell, we use the generating techniques for white, brown and 1/f melody described by Gardner.[6] ACME composes brown melodies by choosing notes that are at most two steps offset from the previous note, and white melodies by choosing notes randomly, without regard to preceding tones. We implement Gardner's algorithm for 1/f melodies, which produces tone sequences that fall somewhere between white and brown. At this point ACME also assigns durations to the notes, picking lengths randomly from eighth to whole note.

Because the first note in a melodic cell can be important in determining the successive notes, we had ACME search a corpus of popular country music to determine the likelihood of a phrase starting with each scale tone. Phrases tended to begin on the tonic or dominant, and ACME uses its analysis of starting note frequencies to stochastically pick the beginning tones for melodic cells.

After assembling a melodic cell of 1-4 notes as described above, ACME follows the rules described in [3] to combine the melodic cells into motives:

- Repetition: Repeat the melodic cell exactly.
- Translation: Repeat the melodic cell at higher or lower intervals.
- Division: Repeat a portion of the melodic cell.
- Modification: Repeat a portion of the cell with modified intervals.
- Introduction: Begin a novel melodic cell.

ACME chooses a series these operations stochastically: for this prototype the each rule has equal likelihood of being applied when extending a melodic cell. Adjustment of these probabilities could be one way to improve generation in future prototypes, based on feedback from the melody evaluation.

The series of modified melodic cells forms a motive. The same production rules are then applied to motives to form a phrase. This simple method of modifying and repeating the same or novel sections seems to follow the conventions of popular country music. For example, the phrase shown in Figure 3 from a popular country song written by Taylor Swift suggests a similar construction pattern. Swift's motive repeats a two-note melodic cell twice, then repeats a portion of the cell twice. This motive, M, is stated, then repeated twice again slightly modified form, M'. The phrase finishes by introducing a new motive, R, which resolves to the tonic.

After creating a phrase, ACME repeats and modifies the phrase using the same production rules outlined in of [3], to generate a period. ACME concatenates these periods together to function as the verse, bridge or chorus of a song.

ACME's system of melody generation is basic, and there are many established songwriting rubrics that might improve ACME's compositions. From Fux's rule that after a long jump

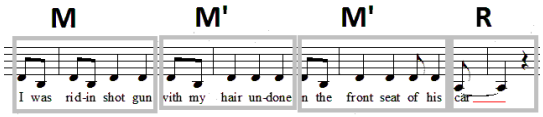


Fig. 3: Melodic Motives.

in one direction melodies ought to take a small step in the other direction [5] to Paul Simon's advice to construct the bridge from notes used infrequently in the verse and chorus [14], there are many heuristics we could incorporate into ACME's generation system. These rubrics suggest that context is important in producing quality melodies, and so a weakness of the context-free grammar is that it is unable to capture interactions between song sections that determine in some way the quality of the melody.

A context-sensitive grammar might produce better average quality, especially when combined with other machine leaning techniques, such as weight search. But it seems unlikely that weights could be found that would generate only good songs. Based on the model of human songwriters, it seems that even a fine-tuned generator will still produce many low-quality songs. For the purposes of providing input to our evaluative system, ACME does the job of creating a large volume of melodies for lead sheets. If the universal search space is the set of all possible melodies, our generation system produces a much smaller set of compositions, because the production is limited to tonal music in a set melody scale, with a fixed length and some repetition of phrases. Further fine-tuning of the generation rules might reduce the search space, at the risk of excluding some portion of the set of country melodies.

5.2 Chord Generation

ACME is able to generate melodies, which are one component of a lead sheet. Lead sheets also have chord symbols for rhythm guitar accompaniment that are used by the arranger to guide creation of harmony parts. A chord is a set of notes that sound at the same time, often strummed on guitar or played on a keyboard. Each chord contains notes from a scale that may be different from the melody scale. Generally a chord is defined by three tones, the chord triad.

Chord changes usually occur at the beginning of measures [12], and tend to draw from scales close to the melody scale on the circle of fifths. [3] We used these two assumptions to simplify ACME's chord generation task: we only allow chord changes on the first beat of a measure, and use only major and minor keys adjacent to the melody scale on the circle of fifths. So for a song in the key of C major, our palette of available chords is C, F, and G major, along with their relative minors A, D and E minor.

Although ACME is able to narrow down the set of possible chords for a phrase and their potential positions, arranging the chord changes in order is a more difficult problem. Texts on harmonizing tend towards subjective evaluation, although there are some common themes. The relation of chords of the sequence of chords is important: Citron suggests beginning with the tonic and ending with the dominant. [3] Piston stresses that chords should change from measure to measure, but not always, and that the distance between keys should be a factor in choosing chords. [12]

The relation between notes in the measure and the associated chord is also important. Chord tones that are dissonant with the notes of the melody are discouraged, and melody notes that are part of the chord triad are encouraged. [12] Agreement of chord tones with longer melody notes and notes at the start of the measure also are considered significant. [1]

Although we lack an algorithm for harmonizing an existing melody, we can describe features that might play a part in determining the suitability of a particular chord accompaniment. The features describe both the relation of notes to their accompanying chord change, and the relation of chord changes to each other:

- Triad Ratio: The ratio of notes governed by the chord change that are part of the chord triad.
- Scale Ratio: Ratio of notes played during the chord that are tones in the chord scale.
- Nonharmonic Ratio: Fraction of notes that are passing tones in the accompanying chord scale.
- Accidental Tones: Ratio of notes that are not members of the chord scale.
- Key Centeredness: 1 if the phrase starts with the tonic chord, 0 otherwise.
- Resolution: 1 if the phrase ends on the dominant chord, 0 otherwise.
- Freshness: Ratio of measures that contain change in chord.
- Mode Freshness: Ratio of chord changes that are different in mode from the previous chord.
- Harmonic Movement: Average distance in fifths between successive chords.

Given a set of candidate chord changes for a melody, ACME can calculate these features for each measure. If we could determine a set of weights to assign to the features, ACME could use that heuristic to rank its own chord changes.

We used a genetic algorithm (GA) to find weights for our chord sequencing heuristic. As training data we used a set of MusicXML lead sheets of contemporary country songs obtained from Wikifonia. ACME created a population of individuals initialized with randomized starting weights, and calculated the fitness of each individual as the linear combination of weights and features each generation. We considered the chord accompaniment in the training set to have an optimal chord fitness of 100, and defined the fitness of individuals as the squared difference of their ranking of the chord changes from 100. Using a population of 1000 individuals for 1000 generations the GA kept the fittest 2/3 each generation, and randomly picked 10% of the remaining individuals for crossover and re-stocked the rest of the popu-

lation with mutation.

After 400 generations the GA produced individuals that had a minimal error against the training data. Figure 4 is a chart of the average and best fitness over 1000 generations.

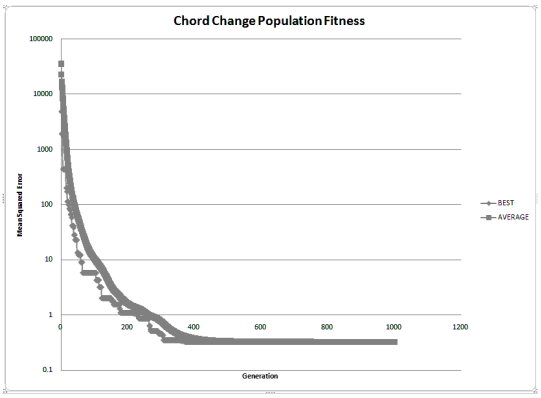


Fig. 4: Chord Weight Population Fitness.

Using the weights of the most-fit individual from the GA as a heuristic, ACME is able to score the fitness of a chord accompaniment to a melody. Given a melodic phrase, ACME generates a set of permutations of chords for the melody. Then, ACME ranks the chord accompaniments and chooses the chord progression with the highest score. Using this method ACME generates chord accompaniments that sound pleasing, while the lowest scoring accompaniments are jarring in their dissonance. As part of the next iteration of ACME we plan to play a selection of low and high scoring chord accompaniments to human listeners and see if their judgements agree with the system's heuristic.

5.3 Reading and Writing MusicXML

MusicXML is an XML-based music file format, introduced in 2004. MusicXML was an attractive format compared to MIDI or PDF because MusicXML is widely supported, can be stored as human-readable text, and appeared to have a reasonable set of examples available online. Free players to edit and export MIDI from MusicXML are also available. The main disadvantage of MusicXML for ACME is that contemporary country songs are generally not available for free. Creating MusicXML samples involved typing in songs by hand from sheet music.

Coding a parser for MusicXML was not difficult, as it is an XML format and C# contains robust classes for navigating XML. We created packages to read and write MusicXML and to allow ACME to save its generated lead sheets as MusicXML files.

5.4 Basic Feature Extraction From MusicXML

Humans, even without formal training, have a strong innate sensibility for music, and can make value judgements on the quality and structure of music. Most people can recognize a variety of musical features: repeated motives, song structure, dissonance, and the ineffable "catchiness" of well-crafted songs.

ACME is able to extract a set of features from a melody stored as MusicXML, and uses those features to emulate a human's ability to make judgements about music. The feature set uses 11 features described in [13] with additional features based on [3], for a total of 80 features. The 11 features that follow from [13] include melodic features, tonality features, melodic contour features, and rhythmic features:

- Pitch Variety: Ratio of distinct pitches to notes.
- Pitch Range: The difference in half-steps between the highest and lowest tones in the melody.
- Key Centeredness: Ratio of dominant or tonic notes (important tones for asserting the melody's key).
- Dissonant Intervals: Ratio of notes whose preceding note is seven semitones distant (intervals of a seventh are dissonant).
- Contour Direction: Overall trend of the melody to rise or fall.
- Contour Stability: Tendency of the melody to continue moving in the same direction.
- Brownian Steps: Ratio of intervals that are a single scale step.
- Leap Restraint: Ratio of large steps that are followed be a small step.
- Note Density: Average number of notes per beat.
- Rest Density: Proportion of beats that are silent.
- Repeated Duration: Proportion of notes that have the same duration as the previous note.

5.4.1 Feature Extraction From Motives

The features described in the previous section all provide aggregate information about the melody as a whole. But, as illustrated in Figure 3, there can be motives in country music that repeat and develop the theme of the melody. The characteristic riff or hook of a song can be recognized by a human listener, and ACME is also able to pick out features from the prominent motive of the melody. ACME finds the most commonly repeated series of notes that recur in the melody: the main motive. By extracting the 11 features described earlier separately for the notes of the motive, ACME has information about the range, direction, variety, and other features of the main motive. In addition, ACME also creates features for the length and number of repetitions of the motive.

Songs can have different motives in different song sections, and motives may not be repeated verbatim. The ability to extract multiple motives and be flexible in recognizing variations would certainly produce a richer feature set. But at present the system has at least some insight into some features of the most reiterated motive.

5.4.2 Feature Extraction From Note Frequency

The base feature set contains one feature, *Key Centeredness*, which describes the combined proportion of the I and V tones in the melody. ACME extends this feature to provide individual proportions for all 7 tones in the melody scale. These 7 scaled values are added to the existing feature set. Having calculated these values, the system is able to create a visualization of the usage of scale tones in the melody. By laying out a histogram of the frequency of note usage, ACME can create human-readable data about the melody during its feature extraction. Figure 5 is a tone frequency histogram generated by ACME during feature extraction.

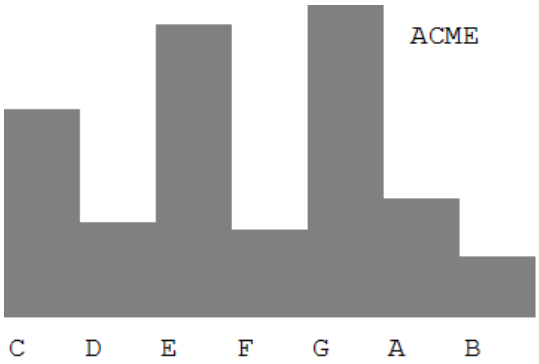


Fig. 5: Tone Frequency Histogram.

5.4.3 Feature Extraction From Note Transition Model

The likelihood of moving from any single scale tone to another can be represented by a 1st-order Markov model. Some of the earliest attempts at algorithmic composition relied on solely Markov models to generate melodies in the style of a corpus of music [11]. To make use of these features, ACME creates a 1st-order Markov model of the scale tone transitions in a melody. The generation system composes using major and minor scales of seven tones, so a 1st-order Markov model of the note transitions has 49 values, which ACME adds to the existing feature set. When these features are plotted graphically, they provide a visual representation of likelihood of transitions from scale tones to other scale tones in the melody. Figure 6 is an example image plotted by the system of the Markov transition features of a generated song. Darker squares represent more common transitions, while lighter squares represent less frequent transitions. The most common transition in the melody represented in Figure 6 is the transition from F# to E.

5.5 Training The Evaluation Neural Network

ACME is capable of extracting features from training examples to use as input for a neural network. For training data, we used a set of several dozen country music songs in MusicXML format. Since the goal of the system is to create songs that might be successful on the radio, ACME uses the

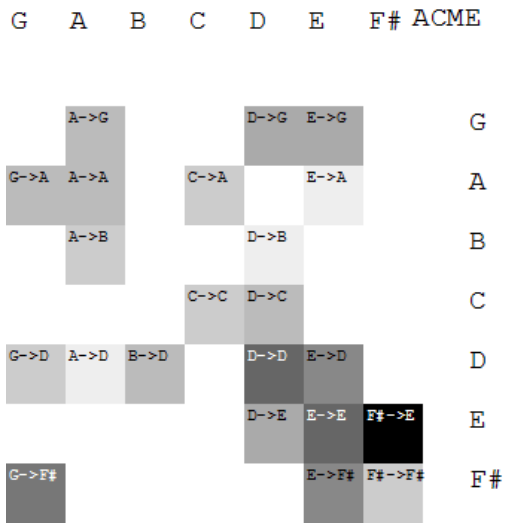


Fig. 6: Note Transition Visualization.

song's Billboard chart ranking (1 being the best and 100 the worst ranking) as the expected value of the examples. We had some difficulty in obtaining contemporary country music lead sheets. There is no shortage of lead sheets, but they are not generally available for free. We were able to find some lead sheets on Wikifonia.org, and entered more manually using a score editor.

To provide negative examples, the system created duplicate, degraded versions of the positive training examples. ACME randomly nudged intervals, reordered notes in measures, split and combined notes, and inserted and deleted rests to existing melodies. These modified examples sound undeniably execrable, and form our set of negative examples. The system assigns a very poor expected value of 1000 to these negative examples.

Given a set of labelled training examples and the ability to extract a set of features, ACME is capable of training a backpropagation neural network. After some experimentation, we obtained satisfactory results from a fully-connected network with 80 neurons in the input layer (one for each of the the 80 features), two hidden layers of 40 neurons each, and an output layer of one neuron. Figure 7 shows the reduction in mean squared error over 300 iterations over the training examples.

5.6 Evaluation of Generated Songs

ACME can generate a few thousand songs per hour, and evaluation with the neural network can keep pace with generation. So ideally, we could sort through 100,000 original songs in a few days and select the best-ranked melody. Given our simple generation algorithm, we would expect that the majority of generated lead sheets should rank poorly. As expected, the neural network does classify the majority of the output as poor. In one run ACME generated and evaluated 5000

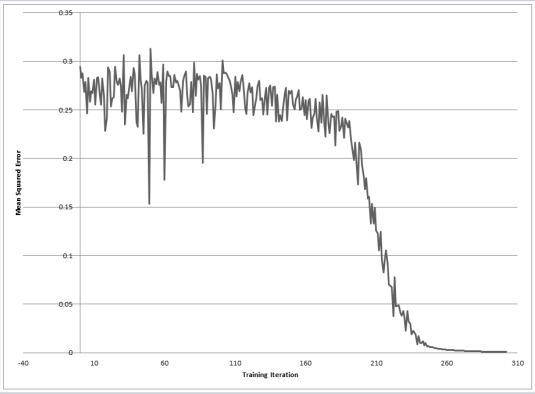


Fig. 7: Mean Squared Error Over 300 Iterations.

lead sheets, with the classification representing the putative chart position of the generated song: 0 being the best possible ranking and 1 the poorest. Figure 8 shows the ranking for the 5000 songs, grouped by ranking. Most of the generated melodies fall into the poorest category, while only a small number garner a high rating.

To our ears the poorly rated melodies do not sound pleasing, and although the highly ranked melodies are noticeably better, the best ranked melodies are not of exceptional quality. The high ranked songs are not of equal quality to those produced by a human expert, so the evaluation system could use improvement in discrimination. More features might be needed, and it is also not clear that a neural network is the best choice for evaluation of generated melodies. An alternate technique, such as Bayesian learning, might produce better results and could help illuminate which features or combination of features are most critical for judging the popularity of country music.

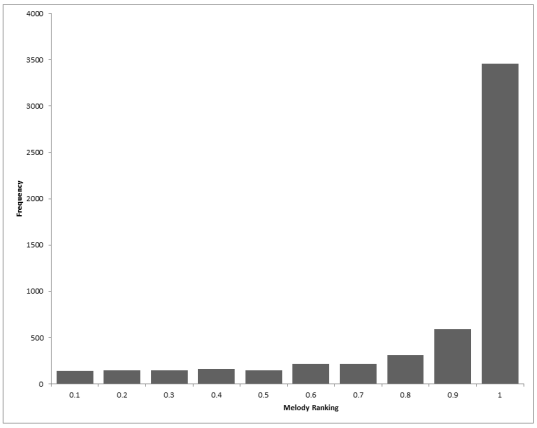


Fig. 8: Generated Song Rankings.

6. Conclusions and Future Work

ACME's results demonstrate that it is possible to generate a large volume of novel melodies and use existing machine learning techniques to evaluate and select those that are "better". From this proof of concept we plan to consider a variety of techniques to improve our results. We will consider adjusting the rules and learning weights for the stochastic context-free grammar to increase the proportion of higher-quality melodies. We plan to investigate alternative generation techniques, such as Markov Models, Bayesian Networks, or a context-sensitive grammar. Creation of such models will of course require a larger database of popular country songs. A larger database can also improve the reliability of training the evaluation component.

Even with a feature set of size 80 the system still is lacking insight into overall song structure (such as the location and arrangement of the verse, chorus and bridge), motives other than the single most common one, and information about the variation and development of motives. So, there seems to be room for improvement in feature extraction. It is likely that not all existing features that are relevant to a song's popularity are known, so we will investigate the possibility of automatically discovering new features.

As the system becomes more mature, we intend to assemble a panel of human listeners to determine if their rankings of generated melodies agree with those of alternative evaluation architectures. We also intend to run the system for an extended period to see if the generator is able to produce a work of high quality. Although our work is at an early stage, we are encouraged that ACME might be able to recognize a musical gem inside a heap of low-scoring rubble.

References

[1] Musicarta.com, *Songwriting Techniques*, http://www.musicarta.com/songwriting-techniques.html , 2011.
[2] Noam Chomsky, *Three Models for the Description of Language*, IRE Transactions on Information Theory, 1956.
[3] Stephen Citron, *Songwriting: A Complete Guide to the Craft*, William Morrow and Company, New York, 1985.
[4] Ada Lovelace, *Sketch of the Analytical Engine Invented by Charles Babbage, Esq*, Scientific Memoirs, Vol 3 (1842)
[5] Johann Joseph Fux, *The Study of Counterpoint*, Translated and edited by Alfred Mann, Norton, New York, 1971.
[6] Martin Gardner, *White and Brown Music, Fractal Curves and 1/f Fluctuations*, Scientific American, 234(4): 16-32 April 1978.
[7] Tom T. Hall, *How I Write Songs*, Chappell Music Company, New York, 1976.
[8] Roman Klinger, Gunter Rudolf, *Evolutionary Composition of Music with Learned Melody Evaluation*, Proceedings of the 5th WSEAS International Conference on Computational Intelligence, Man-Machine Systems and Cybernetics, 2006.
[9] Fred Lerdahl, Ray Jackendoff, *A Generative Theory of Tonal Music*, The MIT Press, 1982.
[10] Gerhard Nierhaus, *Algorithmic Composition*, SpringerWein, New York, 2009.
[11] Henry F Olson, *Music, Physics, and Engineering*, Dover Publications, New York, 1967.
[12] Walter Piston, *Harmony*, W. W. Norton & Company, New York, 1962.
[13] Micheal Towsey, Andrew Brown, Susan Wright, and Joachim Diederich *Towards Melodic Extension Using Genetic Algorithms*, Educational Technology and Society, 4(2) 2001.
[14] Paul Zollo, *Songwriters on Songwriting*, Writer's Digest Books, Cincinnati, 1991.

Meta-learning based Optimization of Social Feature Extraction Inference System

Wladimir Ormazábal, Tomás Arredondo, *Member, IEEE*, Werner Creixell, *Member, IEEE*, Sebastián Contreras and Patricio Olivares

***Abstract*— In this article we propose a generic framework for Meta Learning (MTL) of parameters and models used in the optimization of inference systems. This generic optimization approach can be applied to any hypothesis or learner as long as the optimizer can support the parameters and model values required. This proposal is evaluated classifying the position and type of users in a social database. The obtained results are contrasted with other classification methods and with manually optimized inference systems.**

I. Introduction

A large percentage of the world is carrying mobile phone devices with computing capabilities [1]. There have been numerous projects that have looked to take advantage of the intrinsic co-location characteristic of owners and their cellular telephones in order to determine more about human behavior. Early studies into the social behavior of people using this technology have tried to learn more about their mobility patterns.

In [2] the trajectory of 100,000 anonymized mobile users for a period of six months were studied. One shortcoming that was found is that the precision of extracting mobility data from Call Data Records (CDRs) provided by telecommunication companies is determined by the cellular tower coverage area. This precision varies widely, from less than 1 km^2 in highly populated areas to more than 4 km^2 in others. The CDR data is also not periodically taken, normally a CDR can give an estimated location only when the phone is being used in a call.

Eagle et al. [3] solved this problem by logging the cellular tower ID data in the phone itself. Clustering techniques were used to infer relevant location, for mobility prediction with over 90% of accuracy. In the work of Ashbrok et al. [4], location data from a single user carrying a GPS receiver was logged during four months. They proposed a method based on clustering to identify meaningful locations among the collected data, and a Markovian model for the user "transitions" between the locations.

The "Reality Mining" project presented in [5] collected various data types from cell phones in a trial of 94 users over a nine month period. This data set was used to estimate the user's position, by cell tower ID patterns, user's daily routine and social networking. The "Reality Mining" project data was used in the work of [6] to further investigate the structure in the data set. Other work that has also used the "Reality Mining" data set is found in [7], the authors seek to discover the daily routine of users from the data set. To do this, a framework was constructed from two Hierarchical Bayesian models based on routine as given by location: Latent Dirichlet Allocation and Author Topic Model.

In our research, we propose a meta-learning (MTL) based optimizer of a social classifier used to determine a particular user position and type. We propose a conceptual structure of MTL centered upon an analysis of the factors that influence the learning process, such as the data available, training method parameters and classifier structural factors (i.e. model configuration).

The rest of this work is organized as follows. In Section II, we analyze prior investigations in meta-learning and we present the new conceptual model. In Section III, we show the experiments that were performed and we give results. Finally, Section IV elaborates the conclusions from this study and some proposals for future work.

II. Meta-learning Model

In general there is no superior learning or inference method that is superior to others in all applications [8], [9], that is why it is of considerable interest to be able to obtain a good inference system as a result of a learning process independent of the problem domain. The factors that affect these changes in behavior as well as the search for processes and methods that improve the process of learning is denoted as "meta-learning" [10], [11], [12], [13].

A learner can be defined as an inference system that is trained to generate an output based on certain inputs. The desired input-output relationship can be seen as a function (i.e. a hypothesis) and the learning process seeks to reduce the error between the actual and the desired output of the system [14], [15].

A weak learner is defined as a hypothesis that has a performance which is a bit better than random prediction [16], [17]. This performance can be considered as the error

Wladimir Ormazábal is with the Department of Electronic Engineering, Universidad Técnica Federico Santa María, Valparaíso, Chile. (phone: 56-32-2654212; fax: 56-32-2797469; email: wladimir.ormazabal@gmail.com).

Tomás Arredondo is with the Department of Electronic Engineering, Universidad Técnica Federico Santa María, Valparaíso, Chile. (phone: 56-32-2654212; fax: 56-32-2797469; email: tomas.arredondo@usm.cl).

Werner Creixell is with the Department of Electronic Engineering, Universidad Técnica Federico Santa María, Valparaíso, Chile. (phone: 56-32-2654758; fax: 56-32-2797469; email: werner.creixell@usm.cl). He is also visiting researcher with CSIS at The University of Tokyo, Japan.

Sebastián Contreras is with the Department of Electronic Engineering, Universidad Técnica Federico Santa María, Valparaíso, Chile. (phone: 56-32-2654758; fax: 56-32-2797469; email: cba.contrerasm@gmail.com).

Patricio Olivares is with the Department of Electronic Engineering, Universidad Técnica Federico Santa María, Valparaíso, Chile. (phone: 56-32-2654758; fax: 56-32-2797469; email: paolivaresr@gmail.com).

obtained between the desired and obtained results or the percentage of errors over all test cases [18]. Design decisions regarding weak learners can consider the implemented model as well as the learning method and the parameters used by this method all of which influence the learning performance [19], [20], [21], [18], [22].

It is possible to find diverse approaches that have been categorized as MTL [11], [12], all these approaches eventually aim to obtain the best result from the learning process. Some have centered in the study of the nature of the data as in the number of characteristics to analyze, the classes being classified or the amount of training data amongst others [11], [12] (e.g. Landmarking [23]); others have applied combinations of base learners as a means to improve the learning outcome (e.g. Bagging [24], Boosting [25], Stacked Generalization [26]).

Meta-optimization [27] is another approach that focuses on finding parameters for a learning method that produces the best results. Typically this search is based on optimization strategies such as: backpropagation [28], Genetic Algorithms (GA) optimization of Fuzzy Inference System (FIS) functions [20] and fuzzy rule sets [21] or Particle Swarm Optimization (PSO) of artificial neural network (ANN) parameters [18]. The work developed in [18] focuses on the modification of learning parameters using an algorithm called "Local Unimodal Sampling" (LUS) toward the search and selection of PSO parameters which in turn is used to find optimal weights for a neural network. A more complete approach can be seen in [22] where ANN model design and the training methods being utilized are incorporated into the optimization process.

The majority of previous works in MTL [19], [20], [21], [18], consider certain aspects of interest (e.g. learning parameters or learning model structural elements). To the best of our knowledge, only [22] approaches both parametric aspects and model configuration but only in a context specific manner.

Our current proposal is a general conceptual framework that generalizes the concept of MTL independent of the inference system (i.e. learner or weak learner) and the algorithm being used to optimize it (e.g. backpropagation, genetic algorithms, etc).

A. Training and validation data definition

There are diverse possible data sets, given a determined context or problem of interest. Considering one of these possible sets, $\mathbf{D}$, as a set of N elements:

$$\mathbf{D} = \{d_1, d_2, \ldots, d_N\}, |\mathbf{D}| = N \tag{1}$$

Where:

- $d_i = (I_i, O_i)$, with $1 \leq i \leq N$.
- I_i: Input values.
- O_i: Output values.
- $|X|$: Cardinality of the set X.

Following [14], it is possible to divide the set $\mathbf{D}$ in r disjoint subsets T:

$$\mathbf{D} = T_1 \cup T_2 ... \cup T_r, \text{ with } T_i \cap T_j = \emptyset, \quad \forall i, j \in \{1, 2, \ldots, r\}, \ i \neq j \tag{2}$$

Starting from these r subsets we can generate a training set (S_i) and a test set (T_i). T_i corresponds to one of the r subsets, and S_i is the union of the $r-1$ remaining subsets. We thus define a training set S_i as:

$$S_i = \mathbf{D} - T_i, \text{ with } \mathbf{D} = \{T_1, T_2, \ldots, T_i, \ldots, T_r\} \tag{3}$$

Considering expression (2) we establish that:

$$\mathbf{D} = T_i \cup S_i \ , \ \forall \ i \in \{1, 2, \ldots, r\} \tag{4}$$

B. Training methods and inference system model

Extending the idea of learner (i.e. inference system) training used in [14], we define L_k as a learner formed by the pair (learning model,learning method) as:

$$L_k = (mod_i, tr_j), \quad \text{with} \quad mod_i \in \mathbf{MOD}, \ tr_j \in \mathbf{TR} \tag{5}$$

Where $\mathbf{MOD}$ is the set of possible learners (e.g. neural network, fuzzy inference system, etc.) and $\mathbf{TR}$ is the set of possible learning methods that can be used to train those models (e.g. backpropagation, genetic algorithm, etc.).

Consider $\mathbf{L}$ as the set of possible different learners, that is the valid combinations (mod_i, tr_j) of learning models and training methods:

$$\mathbf{L} = \{L_1, L_2, \ldots, L_k, \ldots\}, \quad k \in \mathbb{N} \tag{6}$$

Considering expressions (5) and (6), as well as the aforementioned restriction it is possible to assume that:

$$|\mathbf{MOD} \times \mathbf{TR}| \geq |L| \tag{7}$$

For every L_k we define a vector $\mathbf{p}$, that contains all parameters of the training method being used by L_k. We let $\mathbf{P_k}$ be the set of all γ possible $\mathbf{p}$ vectors:

$$\forall L_k \ \exists \text{ a } \mathbf{P_k} = \{\mathbf{p_{k,1}}, \mathbf{p_{k,2}}, \ldots, \mathbf{p_{k,\gamma}}\} \tag{8}$$

In a similar manner, we define the set $\mathbf{M_k}$ of all θ possible inference system model configurations, each one of these represented by a vector $\mathbf{m}$:

$$\forall L_k \ \exists \text{ a } \mathbf{M_k} = \{\mathbf{m_{k,1}}, \mathbf{m_{k,2}}, \ldots, \mathbf{m_{k,\theta}}\} \tag{9}$$

C. Hypothesis

We understand a hypothesis h, as an instance of an inference system type (e.g. a neural network learner) after undergoing the learning process [14]. There is an infinite set of hypotheses h associated to one of the learners in $\mathbf{L}$. For example, $\mathbf{H_A}$ is the set of possible hypotheses generated from learner L_A:

$$L_A \rightarrow \mathbf{H_A} \tag{10}$$

The hypothesis generated depends on the training data that is utilized, hence we can establish that a hypothesis h, not only depends on the learner L_A, but also on the training

set used S_i. We define $\mathbf{H_A}(S_i)$ as the set of hypotheses generated from the 2-tuple defined by L_A and training data set S_i:

$$(S_i, L_A) \rightarrow \mathbf{H_A}\,(S_i) \text{ with } \mathbf{H_A}(S_i) \subseteq \mathbf{H_A} \quad (11)$$

As seen in (8), since the learning method uses a variety of parameters represented by $\mathbf{P_A}$. The hypothesis generated will depend on these parameters as well. For example, for a learner L_A, we have a vector $\mathbf{p_{A,l}}$ (belonging to the set $\mathbf{P_A}$) with all parameter values used by that learner. In a similar way as before, it is possible to define the set of hypotheses $\mathbf{H_A}(\mathbf{p_{A,l}})$ generated from learner L_A with the parameters used by the learning method (i.e. vector $\mathbf{p_{A,l}}$):

$$(L_A, \mathbf{p_{A,l}}) \rightarrow \mathbf{H_A}(\mathbf{p_{A,l}}) \text{ where } \mathbf{p_{A,l}} \in \mathbf{P_A} \text{ and } \mathbf{H_A}(\mathbf{p_{A,l}}) \subseteq \mathbf{H_A} \quad (12)$$

Another factor to consider is the configuration of the model that is being trained represented by $\mathbf{M_A}$, and which much like the previous factors also helps determine the final hypothesis that is obtained. For example, a vector $\mathbf{m_{A,j}}$ that represents a possible model configuration for a learner L_A, will affect the final hypothesis h ($\mathbf{m_{A,j}}$ belongs to $\mathbf{M_A}$).

We also define $\mathbf{H_A}(\mathbf{m_{A,j}})$ as the set of hypotheses generated starting from L_A with a specific model configuration $\mathbf{m_{A,j}}$:

$$(L_A, \mathbf{m_{A,j}}) \rightarrow \mathbf{H_A}(\mathbf{m_{A,j}}) \text{ where } \mathbf{m_{A,j}} \in \mathbf{M_A} \text{and } \mathbf{H_A}(\mathbf{m_{A,j}}) \subseteq \mathbf{H_A} \quad (13)$$

Finally, considering all factors previously mentioned it is possible to establish that a hypothesis h depends on the learner L_A being used, on the training data (S_i), on the model being used ($\mathbf{M_A}$), and on the parameter values of the training method being used ($\mathbf{P_A}$).

We define $h_A^{\beta}(\mathbf{p_{A,l}}, \mathbf{m_{A,j}}, S_i)$ as the hypothesis generated from L_A, with a specific model configuration $\mathbf{m_{A,j}}$, with specific parameters $\mathbf{p_{A,l}}$, and trained with data set S_i (the β value is an index to denote the different hypotheses that can be generated):

$$h_A^{\beta}(\mathbf{p_{A,l}}, \mathbf{m_{A,j}}, S_i) = \mathbf{H_A}(S_i) \cap \mathbf{H_A}(\mathbf{p_{A,l}}) \cap \mathbf{H_A}(M_{A,j}) \quad (14)$$

Beginning with a set of data $\mathbf{D}$ we generate different training sets S_i, also for learner type L_A there are different possible values for $\mathbf{p_{A,l}}$ and $\mathbf{m_{A,j}}$. Hence, starting with $\mathbf{D}$ and L_A we generate diverse hypotheses in the set $\mathbf{H_{A,D}}$:

$$(\mathbf{D}, L_A) \rightarrow \mathbf{H_{A,D}} = \left\{h_A^1, h_A^2, \ldots, h_A^{(\theta \cdot \gamma \cdot r)}\right\} \quad (15)$$

We define $\mathbf{H_A^{l,j}}(\mathbf{p_{A,l}}, \mathbf{m_{A,j}}, \mathbf{D})$ as the set of all hypotheses generated using vectors $\mathbf{p_{A,l}}$ and $\mathbf{m_{A,j}}$ while trained with S_i $(1 \leq i \leq r)$ obtained from $\mathbf{D}$:

$$\mathbf{H_A^{l,j}}(\mathbf{p_{A,l}}, \mathbf{m_{A,j}}, \mathbf{D}) = \left\{h_A^1(\mathbf{p_{A,l}}, \mathbf{m_{A,j}}, S_1), \ldots, h_A^r(\mathbf{p_{A,l}}, \mathbf{m_{A,j}}, S_r)\right\} \quad (16)$$

Each hypothesis belonging to $\mathbf{H_A^{l,j}}(\mathbf{p_{A,l}}, \mathbf{m_{A,j}}, \mathbf{D})$ is obtained from a different training set S_i. We define $h_{A,l,j}$ as a *template* of all hypotheses generated from vectors $\mathbf{p_{A,l}}$ and $\mathbf{m_{A,j}}$ belonging to the set $\mathbf{H_A^{l,j}}(\mathbf{p_{A,l}}, \mathbf{m_{A,j}}, \mathbf{D})$:

$$\forall\, h_A^{\beta}(\mathbf{p_{A,l}}, \mathbf{m_{A,j}}, S_i) \in \mathbf{H_A^{l,j}}(\mathbf{p_{A,l}}, \mathbf{m_{A,j}}, \mathbf{D})\ \ \exists!\ \ h_{A,l,j} \quad (17)$$

Starting with expression (17), we define a hypothesis $h_{A,l,j,i}$ as an instance of the template $h_{A,l,j}$ instantiated with the training set S_i. Also, the output for hypothesis $h_{A,l,j,i}$ given an input I_x is defined as: $h_{A,l,j,i}(I_x)$.

$$h_{A,l,j,i} = h_A^{\beta}(\mathbf{p_{A,l}}, \mathbf{m_{A,j}}, S_i) \quad (18)$$

D. Meta-learner

As seen in Fig. 1, the MTL seeks to obtain the best possible values for the parameters vector ($\mathbf{p_{A,l}}$), this process is denoted as *tuning*. It is also tasked with finding the best possible values for the model configuration ($\mathbf{m_{A,j}}$), this process is known as *fitting*. For a particular template, the generated hypotheses allow us to obtain a measure of fitness for the template that instantiated them. The measure of fitness used to evaluate the diverse hypotheses generated is the mean error over r of the test set T_i applied to all hypotheses (i.e. E_{MTL}). This fitness measure allows the MTL to evaluate and select the best combination of $\mathbf{p_{A,l}}$ and $\mathbf{m_{A,j}}$ found for a specific problem.

One important observation is that there is an inherent degree of recursion in meta-learning. Given that the performance of the MTL also depends on model design considerations and on parameters. Therefore, it is possible to define a meta-meta-learner that seeks to optimize the meta-parameters used by the meta-learner to obtain a better classification performance.

$$E_{MTL} = \frac{1}{r}\sum_{1}^{r}\left(1 - \frac{\text{Correctly classified}}{\text{Total cases}}\right) \quad (19)$$

III. Experiments and Results

A. Reality mining dataset

In the Reality Mining project conducted during 2004-2005, 94 subjects volunteered to participate using an application that was preinstalled in their mobile devices. The application logged call history (including SMS and Internet navigation), Bluetooth devices encountered (i.e. in a 5 meter range), cellular tower identifiers which the phone was registered to, and the usage of mobile applications. The users also labeled some relevant places like *home*, *work* and *else*, directly in their cellphones. The location label was then associated to the cell tower to which the cellphone was connected at the time of labeling. We pre-processed the data using an approach presented in [29], [30] for the posterior use of the classifier.

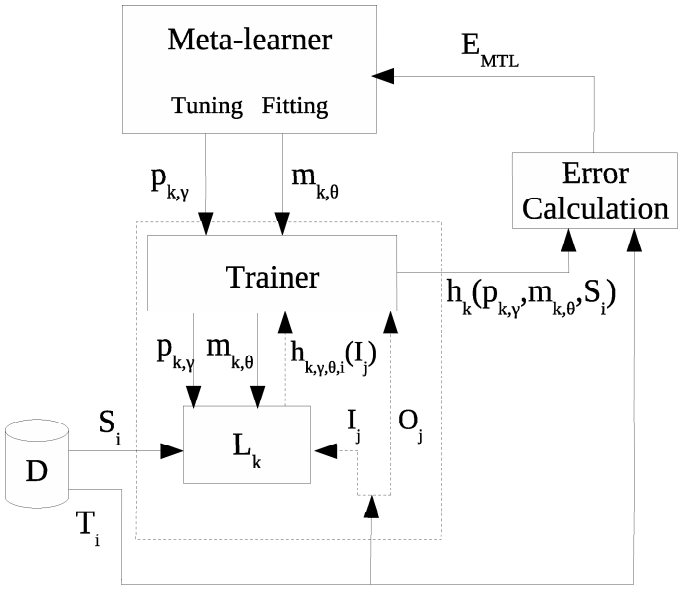


Fig. 1. Inference System Optimizer

TABLE I
MODEL VARIABLE VALUES AND RANGES

Learner	N_i	N_h	N_o
Fixed-NN	4000	1	1
NN1	4000	1 - 30	1
NN2	4000	1 - 30	1

TABLE II
BACKPROPAGATION VARIABLE VALUES AND RANGES

Learner	η	η_{low}	η_{high}	α	d
Fixed-NN	0.001	N.A.	N.A.	0	N.A.
NN1	0 - 0.1	N.A.	N.A.	0 - 0.9	N.A.
NN2	0.3	0.01	0.01 - 0.3	0 - 0.9	20 - 84

B. Neural network learner

As our learner (i.e. L_k), we use a multilayer feed forward artificial neural network with one hidden layer and backpropagation for weight updating:

$$\Delta w_{ij}(n+1) = \eta \cdot \delta_{Pj} o_{Pj} + \alpha \cdot \Delta w_{ij}(n) \qquad (20)$$

We applied two implementations of backpropagation during the process of meta-learning, the first one (NN1) using with a fixed learning rate (η) and the second one (NN2) with a variable learning rate. According to [31], during each learning cycle the initial value of η decreases exponentially with the d factor until reaching the value η_{low}. Upon reaching this value η_{low}, η takes the value of η_{high} and this process repeats:

$$\eta(t) = \eta(t-1) \cdot \exp\left(\frac{\log\left(\frac{\eta_{low}}{\eta_{high}}\right)}{d}\right) \qquad (21)$$

In Tables I and II, neural network (NN) learner variables of interest and their ranges of values are shown. In *Fixed-NN*, we indicate the values for the best NN configuration found during extensive previous manual experimentation (i.e. without meta-learning) for the two locations (Home and Work). In rows labeled *NN1* and *NN2* we show the range of values used by all the meta-learners for fixed and variable learning rates. These values were considered to have the greatest possible impact upon hypothesis generation after a literature review and previous experimentation.

C. Training Methodology

The training process consisted in doing 50 runs of network training and testing with random initial weights as well as random training and test sets. Toward the selection of training and test sets, we have applied cross validation random subsampling [32], in a 70% - 30% proportion respectively.

The fitness of the neural network configuration was calculated as the best average performance over all test runs. This process was repeated for all users in the database.

To avoid overtraining, we utilized T_i to monitor the change in error according to (22). After each training round in case of an increase in error the process is stopped. We restricted the possible maximum number of training iterations to 20000 and the minimum number to 1000.

$$Error_i = \frac{1}{|T_i|} \sum_{x=0}^{|T_i|-1} \frac{1}{2} \left(O_j - h_{A,l,j,i}\left(I_x\right)\right)^2 \qquad (22)$$

D. Meta-learning algorithms

We utilized a genetic algorithm (GA) and Stochastic Hill Climbing (SHC) in order to implement the process of MTL.

1) Genetic Algorithm: The GA modified the number of hidden layer neurons (N_h) as well as backpropagation parameters according to the test case (fixed or dynamic learning rate). The performance of the GA is also associated with a series of parameters and considerations which determine its behavior.

Previous researchers have determined certain reasonable ranges for GA parameters to be used in typical optimization problems [33], [34]. From these, we have chosen the following parameters for the GA: a population of 20 individuals, two point crossover, probability of crossover of 0.95, probability of mutation of 0.01 and 100 iterations. We also used Tournament selection with one Elite individual. In GA-NN1, we used a granularity of 5 bits for N_h, 14 bits for η, and the momentum (α) was also 14 bits. For GA-NN2 we used 5 bits for N_h, 6 bits for η_{high}, α was 12 bits and 6 bits for d.

2) Stochastic Hill Climbing: Another algorithm that was utilized for MTL was Stochastic Hill Climbing (SHC) [35]. This algorithm modified the same parameters as the GA. We used an equivalent number of objective function evaluations in order to enable some means of comparison with the GA. The parameters used are 200 iterations and 10 restarts. In SHC-NN1, we used a granularity of 5 bits for N_h, 14 bits

for η, and α was also 14 bits. For SHC-NN2 we used 5 bits for N_h, 6 bits for η_{high}, α was 12 bits and 6 bits for d.

E. Experiments

We used the same training methodology for all experiments. The first experiment consisted in identifying the user locale: *home*, *work* or *else* for all users. Afterwards we selected the users with the worst performance reached for location classification, in the case of Home it was user 96 (64.11%) and in the case of Work it was user 43 (61.90%). The last experiment, was the classification of the users into a specific profile: *Staff*, *Sloan*, *Undergraduate*, or *Postgraduate*.

F. Results

In Table III, the total average correct % location classification results reached by all classifiers is compared. GA-NNX corresponds to the GA based MTL training NNX and SHC-NNX is the SHC based MTL training NNX. We also include results from other classifiers, these use standard configurations obtained from prior work: the best one obtained through manual optimization (Fixed-NN), Support Vector Machines (SVM) and Naive Bayes (NB).

Table IV and V indicates the best parameters and configuration values that were found by MTL. Table VI corresponds to the total average correct % user profile classification obtained by the best MTL classifier (i.e GA-NN1) as contrasted with other classifiers.

TABLE III
FINAL LOCATION CLASSIFICATION RESULTS

Learner	96 - Home	43 - Work
Fixed-NN	64.11 ± 1.28	61.90 ± 2.34
SVM	79.51	71.88
NB	79.51	63.82
GA-NN1	70.69 ± 2.80	83.78 ± 2.373
GA-NN2	68.07 ± 1.65	83.25 ± 1.836
SHC-NN1	67.99 ± 1.46	82.96 ± 1.87
SHC-NN2	67.58 ± 2.18	82.80 ± 1.55

TABLE IV
BEST MODEL VALUES OBTAINED

96-Home	N_i	N_h	N_o
GA-NN1	4000	15	1
GA-NN2	4000	24	1
SHC-NN1	4000	22	1
SHC-NN2	4000	5	1
43-Work	N_i	N_h	N_o
GA-NN1	4000	1	1
GA-NN2	4000	2	1
SHC-NN1	4000	8	1
SHC-NN2	4000	22	1

TABLE V
BEST PARAMETER VALUES OBTAINED

96-Home	η	η_{low}	η_{high}	α	d
GA-NN1	0.0003	N.A.	N.A.	0.0206	N.A.
GA-NN2	0.3000	0.01	0.0744	0.0102	26
SHC-NN1	0.0472	N.A.	N.A.	0.3052	N.A.
SHC-NN2	0.3000	0.01	0.1665	0.0178	21
43-Work	η	η_{low}	η_{high}	α	d
GA-NN1	0.0004	N.A.	N.A.	0.2517	N.A.
GA-NN2	0.3000	0.01	0.0422	0.0191	32
SHC-NN1	0.0543	N.A.	N.A.	0.3849	N.A.
SHC-NN2	0.3000	0.01	0.2815	0.3843	47

TABLE VI
FINAL USER CLASSIFICATION RESULTS

User	SVM	NB	Fixed-NN	GA-NN1
Staff	92.166%	92.169%	92.145%	99.220%
Sloan	75.661%	75.688%	70.547%	93.792%
Undergraduate	68.021%	60.838%	70.547%	88.194%
Postgraduate	79.082%	79.147%	56.045%	88.851%

IV. CONCLUSIONS AND FUTURE WORK

As seen by the results of this study, we observe the benefit of using MTL over more traditional ad hoc manual optimization techniques. The automation of the search of classifier model configuration and learning parameters by means of MTL permits the alleviation of a very time intensive and difficult task. The MTL that was implemented does not require constant supervision or intervention and is very useful in a practical sense since other options typically have a financial cost associated with them [31]. The MTL software developed for this optimization is open source and is freely available [36]. Also, the conceptual model that was introduced is that it is very flexible and allows for a series of possible future areas of investigation while maintaining model consistency.

To the contrary of expectations [31], the utilization of the backpropagation method with a dynamic learning rate does not seem to improve results. When comparing the algorithms utilized by the MTL we observe that the GA obtains configurations with better performance in general than the SHC. This may be related to the problem under study and itself is an area of possible further investigation. When contrasting users, the performance of the MTL based classifiers is equivalent if not a bit better than other well known classifiers such as SVM and NB.

Future work includes investigating other parameters and model values to be optimized, using different meta learning algorithms and evaluating the impact of meta parameter granularity on MTL performance.

ACKNOWLEDGMENTS

This research was financed by the Universidad Técnica Federico Santa María by the DGIP Grants 231021 and 231138. We want to thank Nathan Eagle from MIT for kindly providing the Reality Mining Dataset and the associated documentation and also Ethel Vidal for reviewing this manuscript.

REFERENCES

[1] I. T. Union, "Measuring the information society: The ict development index," Tech. Rep., 2010.

[2] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, Jun. 2008. [Online]. Available: http://dx.doi.org/10.1038/nature06958

[3] N. Eagle, J. A. Quinn, and A. Clauset, "Methodologies for continuous cellular tower data analysis," in *Proceedings of the 7th International Conference on Pervasive Computing*, ser. Pervasive '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 342–353.

[4] D. Ashbrook and T. Starner, "Using gps to learn significant locations and predict movement across multiple users," *Personal Ubiquitous Comput.*, vol. 7, pp. 275–286, October 2003.

[5] N. Eagle, A. Pentland, and D. Lazer, "Inferring Social Network Structure using Mobile Phone Data," *PNAS*, 2007.

[6] N. Eagle and A. Pentland, "Eigenbehaviors: Identifying Structure in Routine," in *Proc. Roy. Soc. A (in submission)*, 2006.

[7] K. Farrahi and D. G. Perez, "What did you do today?: discovering daily routines from large-scale mobile data," in *MM '08: Proceeding of the 16th ACM international conference on Multimedia*. New York, NY, USA: ACM, 2008, pp. 849–852.

[8] R. B. Rao, D. Gordon, and W. Spears, "For every generalization action, is there really an equal and opposite reaction? analysis of the conservation law for generalization performance," in *Proceedings of the Twelfth International Conference on Machine Learning*. Morgan Kaufmann, 1995, pp. 471–479.

[9] D. H. Wolpert, "The existence of a priori distinctions between learning algorithms," *Neural Comput.*, vol. 8, pp. 1391–1420, October 1996.

[10] Tom Schaul and Juergen Schmidhuber, "Metalearning. scholarpedia, 5(6)."

[11] M. L. Anderson and T. Oates, "A review of recent research in metareasoning and metalearning," *AI Magazine*, vol. 28, pp. 7–16, 2007.

[12] R. Vilalta and Y. Drissi, "A perspective view and survey of meta-learning," *Artif. Intell. Rev.*, vol. 18, pp. 77–95, October 2002. [Online]. Available: http://portal.acm.org/citation.cfm?id=606412.606413

[13] R. Vilalta, C. Giraud-carrier, P. Brazdil, and C. Soares, "Using meta-learning to support data mining," 2000.

[14] T. Mitchell, *Machine Learning (Mcgraw-Hill International Edit)*, 1st ed. McGraw-Hill Education (ISE Editions), Oct. 1997.

[15] H. A. Simon, "Why should machines learn?" in *Machine Learning: An Artificial Intelligence Approach*, R. Michalski, J. Carbonnel, and T. Mitchell, Eds. Palo Alto, CA: Tioga, 1983, pp. 25–37.

[16] M. Kearns and L. Valiant, "Cryptographic limitations on learning boolean formulae and finite automata," *J. ACM*, vol. 41, pp. 67–95, January 1994.

[17] R. E. Schapire, "The strength of weak learnability," *Mach. Learn.*, vol. 5, pp. 197–227, July 1990.

[18] M. E. H. Pedersen and et al., "Simplifying particle swarm optimization," 2009.

[19] Y. O. T. Aran O., "An incremental framework based on cross-validation for estimating the architecture of a multilayer perceptron," 2009.

[20] T.-P. Hong, W.-T. Lin, C.-H. Chen, and C.-S. Ouyang, "Learning membership functions in takagi-sugeno fuzzy systems by genetic algorithms," in *Proceedings of the 2009 First Asian Conference on Intelligent Information and Database Systems*. Washington, DC, USA: IEEE Computer Society, 2009, pp. 301–306.

[21] T. V. Arredondo, F. M. Vásquez, D. C. Candel, L. Dombrovskaia, L. Agulló, M. H. Córdova, V. Latorre-Reyes, F. B. Calderón, and M. P. Seeger, "Dynamic penalty based ga for inducing fuzzy inference systems," in *Proceedings of the Congress on pattern recognition 12th Iberoamerican conference on Progress in pattern recognition, image analysis and applications*, ser. CIARP'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 957–966.

[22] A. Abraham, "Meta-learning evolutionary artificial neural networks," *Journal, Elsevier Science, Netherlands*, vol. 56, pp. 1–38, 2003.

[23] H. Bensusan and C. Giraud-Carrier, "Casa batlo is in passeig de gracia or landmarking the expertise space," in *Proceedings of the ECML'2000 workshop on Meta-Learning: Building Automatic Advice Strategies for Model Selection and Method Combination*. ECML'2000, June 2000, pp. 29–47.

[24] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, pp. 123–140, August 1996.

[25] Y. Freund and R. E. Schapire, "Experiments with a New Boosting Algorithm," in *International Conference on Machine Learning*, 1996, pp. 148–156.

[26] D. Wolpert, "Stacked Generalization," *Neural Networks*, vol. 5, pp. 241–259, 1992.

[27] M. E. H. Pedersen and A. J. Chipperfield, "Simplifying particle swarm optimization," *Appl. Soft Comput.*, vol. 10, pp. 618–628, March 2010.

[28] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural Networks, 1989. IJCNN., International Joint Conference on*, Jun. 1989, pp. 593–605.

[29] M. A. Bayir, M. Demirbas, and N. Eagle, "Discovering spatiotemporal mobility profiles of cellphone users," in *WOWMOM*, 2009, pp. 1–9.

[30] C. S. Creixell W., Arredondo V. Tomás., "Meaningful locations from celltower information data," 2010.

[31] D. Watkins, "Clementine's neural networks technical overview. technical report." 1997.

[32] R. R. Bouckaert and E. Frank, "Evaluating the replicability of significance tests for comparing learning algorithms," in *In PAKDD*. Springer, 2004, pp. 3–12.

[33] K. A. De Jong, "An analysis of the behavior of a class of genetic adaptive systems." Ph.D. dissertation, Ann Arbor, MI, USA, 1975.

[34] J. Grefenstette, "Optimization of control parameters for genetic algorithms," *IEEE Trans. Syst. Man Cybern.*, vol. 16, pp. 122–128, January 1986.

[35] A. Juels and M. Wattenberg, "Stochastic hillclimbing as a baseline method for evaluating genetic algorithms," Tech. Rep., 1994.

[36] http://profesores.elo.utfsm.cl/∼tarredondo/simmetalib.html.

A Guided Learning Algorithm for solving the Traveling Salesman Problem

Shubham Shukla and Larry D. Pyeatt
Department of Computer Science, Texas Tech University, Lubbock, TX 79409, USA

Abstract— *We propose a novel algorithm for solving the Traveling Salesman Problem (TSP). Our approach uses Genetic Algorithm as a guiding tool to direct Monte Carlo search towards convergence. This hybrid approach leverages a Genetic Algorithm to generate individuals with highly fit genes and uses that information as a base plan for Monte Carlo search. Our experiments suggest that this two-step Expectation-Maximization approach not only converges faster than either of the two algorithms studied individually, but also arrives at solutions that are closer to optimal.*

Keywords: Action Values, Genetic Algorithms, Monte Carlo, Traveling Salesman Problem

1. Introduction

Suppose you are a logistics services company assigned the task of delivering goods to a set of customers. Given the list of customers and their pair-wise distance from each other, the goal is to minimize operational cost by finding the shortest tour which starts and ends at the depot and serves each customer exactly once. This is a minor variation on what is commonly known as the Traveling Salesman Problem (TSP).

Being an NP-Hard problem, various approximation algorithms for solving the TSP have been widely studied in conventional literature. Two such techniques are Genetic Algorithms (GAs) and On-Policy Monte Carlo (MC).

GAs are search algorithms based on the mechanics of natural selection and natural genetics [1]. The convergence of GA solutions is guaranteed by schemata theory [2]. Though highly effective, GAs involve blind-search through the search space, focused on optimizing the population based on some fitness criteria. Their effectiveness can be further improved by finding patterns within the high-fitness genes and using that knowledge to improve the search.

MC methods are Reinforcement Learning (RL) techniques which require only experience, i.e. sample sequences of states, actions and rewards from on-line or simulated interaction with the environment [3]. They converge through a repeated cycle of policy iteration and policy improvement [4]. MC techniques retain knowledge learned through experience, but learning through MC is slow when compared with other optimization techniques [5].

We propose a fusion of GA and MC algorithms (hence, the name: MG) which leverages the *safety-in-numbers* of GAs and retention of *learned knowledge* of MC. Thus, by being a part of a large population (as is the case with GAs), a high-fitness schemata is less likely to be destroyed. Moreover, by constantly updating the action-values (as is the case with MC), learned knowledge is retained for future generations.

In the following sections, we discuss the mathematical model, implementation and results obtained by using Pure-GA, Pure-MC and MG algorithms. Moreover, we analyze the results to establish that MG algorithm converges to a better solution in lesser time.

2. Mathematical Model of TSP using Pure GA

An efficient encoding of parameters and formulation of a robust fitness function are the key aspects determining the quality of solution obtained by GAs. The following are the implementation details of both of these parameters for Pure-GA algorithm:

2.1 Chromosome encoding for TSP

We use a cyclic-permutation encoding scheme [6], [7], [8] where each allele represents a city and the last allele in the chromosome is assumed to be connected to the first, thereby connecting the cyclic tour. The relative position of each allele represents the sequence in which each city is visited during the tour.

Let C_i be the cities to be visited (including the depot) where $i \in \{1, 2, \ldots n\}$. So, each chromosome can be represented by:

$$X = (X_1, X_2, \ldots X_i, \ldots X_n) \tag{1}$$

For the Traveling Salesman Problem, this encoding scheme represents the tour:

$$X_1 \rightarrow X_2 \rightarrow \ldots \rightarrow X_i \rightarrow \ldots \rightarrow X_n \rightarrow X_1 \tag{2}$$

The following conditions are needed for the validity of the chromosome:

1) $X_i \in C, 1 \leq i \leq n$
2) $X_i \neq X_j, i \neq j, 1 \leq i, j \leq n$
3) $X_i = X_j, i = j, 1 \leq i, j \leq n$

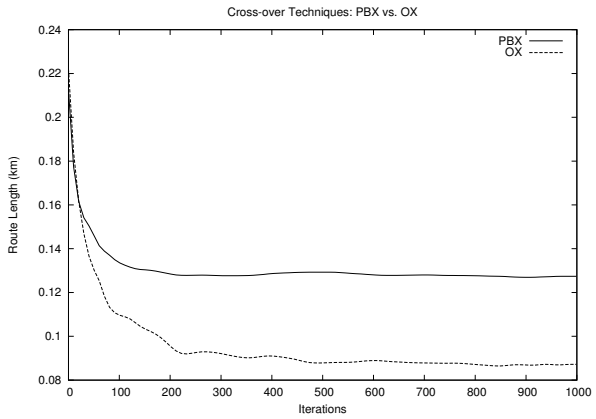


Fig. 1: Crossover: PBX vs. OX

1) Select an initial population P_i randomly.
2) Repeat forever:
   a) Selection:
      i) Select the top 10% population using *elitist* selection.
      ii) Run a biased roulette wheel for the remaining 90% individuals.
   b) Crossover:
      Perform ordered crossover with $p_c = 0.80$.
   c) Mutation:
      Swap two alleles randomly with $p_m = 0.03$.

Fig. 2: Algorithm for Pure-GA for TSP.

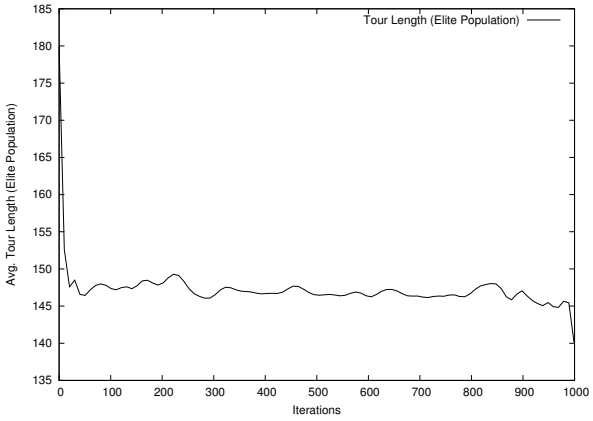


Fig. 3: Pure GA: Convergence of Tour Length

2.2 Fitness Function

The goal of TSP is to minimize the tour length. So for any individual, $(X_1, X_2, \ldots X_i, \ldots X_n)$, in the population, the fitness function is given by:

$$f(X) = \frac{1}{D(X_n, X_1)} + \sum_{k=1}^{n-1} \frac{1}{D(X_k, X_{k+1})} \qquad (3)$$

where $D(X_i, X_j)$ is the distance between the two cities X_i and X_j.

3. Genetic Operators

3.1 Selection Operator

We use an elitist selection scheme where the top 10% of high-fitness chromosomes are retained in the next generation. The remaining 90% are selected using a biased roulette wheel selection [1]. Selecting 10% elite individuals ensures that the solution fitness constantly keeps improving over the generations [9].

3.2 Crossover Technique

Since the TSP is modeled as a permutation problem, simple cross-over at a randomly selected crossover site does not retain ordering information about highly fit schemata.

A crossover operator which respects ordering is therefore an obvious choice. Partially Matched Crossover (PMX) [10] which respects absolute position of an allele in a chromosome, and Ordered Crossover (OX) [11], which respects relative position of an allele are better suited for problems involving permutation than straightforward gene encodings.

We implemented OX and PMX with an asymmetric TSP instance (br17.atsp) to determine which of these two crossover techniques converges to a better solution. $br17.atsp$ comes from the standard set of test problems privided by the University of Heidelberg [12]. Several researchers have routinely used these problems to test their algorithms [13], [14]. An asymmetric TSP instance is a weighted directed graph where the weights between two nodes might not be the same in both the directions.

Results from our experiment are shown in Figure 1. These results suggest that OX outperforms PMX since it retains highly fit allele-sequences in their relative positions for successive generations. For this reason, we use OX for our hybrid implementation.

3.3 Pure GA: Implementation and Results

An implementation of Pure-GA algorithm based on the mathematical model just discussed is shown in Figure 2. The average result from Pure-GA algorithm by averaging over 1000 experiments, is shown in Figure 3.

Though pure-GA algorithm converges drastically in the first 50 iterations before stabilizing to a fairly stable solution, it is nowhere near the best known solution so far for this problem instance (39). The MG algorithm, as we shall see later, converges much more optimally and produces solutions which are even better than the current known benchmark of 39.

4. Mathematical Model of TSP using Pure MC

Monte Carlo techniques are efficient learning techniques for domains with incomplete knowledge of the environment [3]. TSP, being an NP-hard problem requiring no less than exponential time for exact solution, is a suitable candidate for MC learning. To solve a problem using MC techniques, we need to establish a notion of state, action, reward and action values.

4.1 State

State space is defined as the set of all possible stops, including the depot, that a salesman can visit. Relating this to our previous notation, we say that

$$S = \{s_i : s_i \in X\}, \tag{4}$$

where X is the list of stops in the TSP.

4.2 Action

Actions are defined by the set of all possible next stops, from the current state, that have not been visited yet. So, for each state s there is a set of actions $A(s)$, defined as those cities which have not yet been serviced.

4.3 Reward

Since our goal is to minimize the tour length, the reward function should be such that it should assign a higher reward to actions with lower cost. We use the following reward function for our implementation,

$$r(s_i, a_j) = \frac{1}{D(s_i, a_j)}, \tag{5}$$

where $D(s_i, a_j)$ is the distance between the state s_i and the stop at which the salesman arrives by following action a_j.

4.4 Action Values

Action values $Q(s, a)$ express the desirability of going from one state to another by taking an action. It is defined as the average discounted reward obtained at each state in the episode [15].

4.5 Pure-MC: Implementation and Results

An implementation of Pure-MC algorithm based on the mathematical model just discussed is shown in Figure 4 [3]. We implemented an On-Policy version of MC which involves improving the policy that is currently being followed. An ε-greedy policy is used to generate an episode at each iteration.

The average tour length got from On-Policy MC algorithm is shown in Figure 5. The results have been averaged over 1000 trails. As can be seen from Figure 5, the agent learns an increasingly good policy with increasing number

1) Initialize for all $s \in S$ and $a \in A(s)$:
   a) $Q(s, a) \leftarrow$ Uniform Distribution
   b) $Return(s, a) \leftarrow 0$
   c) $\pi \leftarrow$ An arbitrary ε-greedy policy
2) Repeat forever:
   a) Generate an episode using π
   b) For each pair s, a occurring in the episode:
      i) Append $R \leftarrow Return(s, a)$
      ii) $Q(s, a) \leftarrow average(Return(s, a))$
   c) For each state s occurring the episode:
      i) $a^* \leftarrow \arg\max_a Q(s, a)$
      ii) $\pi(s) \leftarrow a^*$

Fig. 4: Pure-MC Algorithm for TSP.

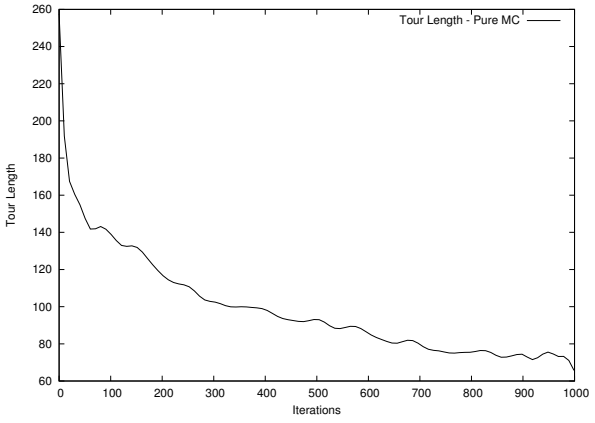


Fig. 5: Pure MC: Convergence of Tour Length

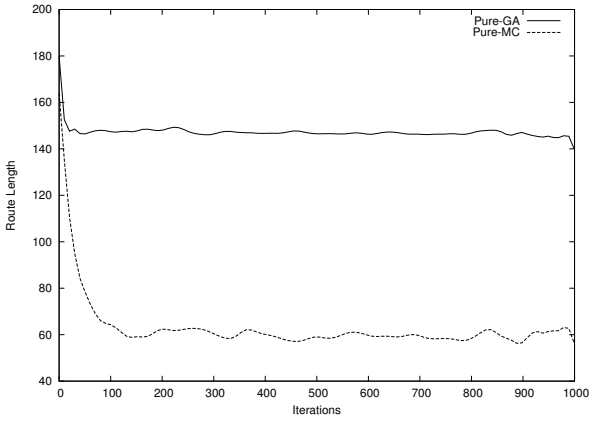


Fig. 6: Convergence Comparison: Pure-MC vs Pure-GA

of iterations. Moreover, comparing the performance of Pure-GA and Pure-MC, as shown in Figure 6, reveals that Pure-MC converges to a much better solution than Pure-GA.

However, neither of these techniques converge to the current-best benchmark of 39. In the next section we discuss the hybrid MG algorithm, which converges to a much better solution in less time than either of the component algorithms alone. Our experiments with MG algorithm suggest that MG is capable of exploring obscure search spaces which were otherwise not explored by Pure-MC and Pure-GA algorithms. This leads to MG finding solutions of route length 27, which are even better than the current-best benchmark.

5. Mathematical Model of TSP using MG Algorithm

The MG algorithm leverages high-fitness sub-structures from genetic search and retains the best policy as a 2-dimensional action-value matrix during the MC phase of the algorithm. During the MC phase, highly fit solutions are further refined by Monte Carlo search. One could consider our approach as a form of Lamarckian evolution, which, although not seen in nature, could be a very efficient search strategy [16].

During the GA phase, MG uses the same reproduction operator as discussed in Pure-GA. We use ordered crossover, since our experiments in ordered crossover, as shown in Figure 1, suggest that OX gives better performance, in general, than PMX.

5.1 Using GA Population for updating action-values

One of the most important concepts used in MG algorithm is the updating of action-values using GA-population. To illustrate how we update action-values using GA population we consider a relatively small problem instance of size 5 (including the depot) for our description. Suppose that the GA population at generation i is:

- A, B, C, D, E
- B, C, A, D, E
- E, D, B, A, C
- A, D, B, C, E

where A, B, C, D, E are the 5 cities (or states S).

Given two adjacent stops (let's say A and B from the first individual), the *desirability* of transition from state A to B will be proportional to the number of strings with A and B adjacent to each other. As GA evolves, each state will have a higher probability to stabilize with the next stop which increases the overall fitness of the population.

Since action-values define the desirability of going from one state to another, we can use this adjacency information to update $Q(s, a)$ at each generation. Mathematically,

$$Q(X_i, X_j) = \frac{count(adj(X_i, X_j))}{pop\_size}, \tag{6}$$

1) Initialize for all $s \in S$ and $a \in A(s)$:
   a) $Q(s, a) \leftarrow$ Uniform Distribution
   b) $Return(s, a) \leftarrow 0$
   c) $\pi \leftarrow$ An arbitrary ε-greedy policy
2) Repeat forever:
   a) Selection:
      i) Select top 10% population using *elitist* selection.
      ii) Run a biased roulette wheel for the remaining 90% individuals.
   b) Crossover:
      i) Perform ordered crossover with $p_c = 0.80$
   c) Generate an episode E using π.
   d) $\forall s_i, a_{i+1} \in E$: Update $Q(s_i, a_{i+1})$ using Equation 6
   e) For each state s appearing in the episode:
      i) $a^* \leftarrow \arg\max_a Q(s_i, s_j)$
      ii) $\pi(s) \leftarrow a^*$
   f) Mutation: Swap two alleles randomly with $p_m = 0.033$.

Fig. 7: MG Algorithm for TSP.

where $X_i, X_j \in S$, $count()$ calculates the number of chromosomes with X_i and X_j adjacent to each other and pop_size is the size of population used for GA. As the GA evolves, each allele (or stop) will be adjacent to its most next stop.

For example, using Equation. 6 for the population above, action-values can be calculated as:

$$Q(A, B) = \frac{2}{4}, \tag{7}$$

and

$$Q(B, C) = \frac{3}{4}. \tag{8}$$

Hence, we use adjacency information from GA population to assign a relative importance of performing an action.

5.2 The MG Algorithm: Implementation and Results

An implementation of MG algorithm based on the update rule just described is shown in Figure 7. MG algorithm starts with the generation of high-fitness chromosomes using selection and crossover operators in steps (2a) and (2b) respectively. Using the adjacency information from this GA-phase, the action values are updated at each iteration (step 2d).

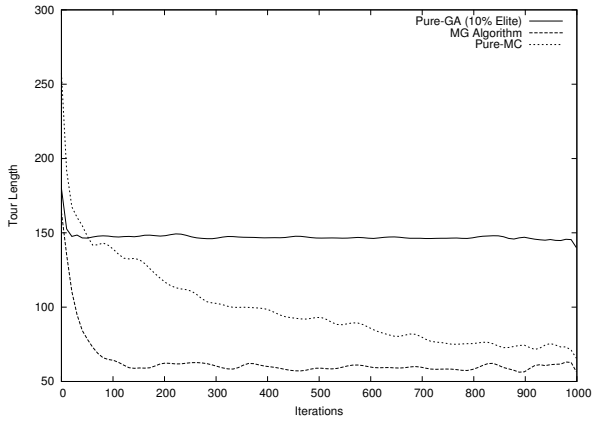


Fig. 8: Comparison: Pure-GA vs. Pure-MC vs. MG

6. Comparison of the Three Methods

We used the br17.atsp asymmetric TSP instance for all our experiments. br17.atsp belongs to a standard TSP library used for validating various TSP algorithms for optimality. The evaluation criteria used for judging the quality of solution got from these three techniques involved the quality of solution and the number of generations required to obtain optimal solution.

As shown in Figure 8, MG algorithm converges to the best solution in about 150 iterations, unlike Pure-MC algorithm which requires about 900 iterations. Although GA does converge to a stable solution in about 30 iterations, but its solution is far from optimal. Moreover, the solution got from MG algorithm is better than MC or GA taken individually.

We also observe that the solutions got from MG algorithm are an improvement over the currently best known solutions. For example, for the br17.atsp problem instance, the best known solution is 39. However, MG algorithm came up with a solution with tour length 27. This suggests that a combination of MC and GA helps explore search spaces which are otherwise left obscure by Pure-MC or Pure-GA techniques [17].

7. Conclusion

We described an effective algorithm for solving TSP using a combination of Monte Carlo and Genetic Algorithms. Our experiments on standard TSP instance suggest that our algorithm gives near optimal solutions. A comparative analysis of this algorithm indicates that this algorithm is more optimal than either of MC or GA techniques studied individually. Thus, by encoding the adjacency information from the population and using it to update the action-values during MC step, we get a powerful algorithm which converges to a better solution in lesser time.

8. Future Work

A possible improvement on MG algorithm can be the use of softmax action selection during MC-step. Using Softmax action selection, instead of ϵ-greedy action selection, can give better performance in the long run.

Also, during the GA-phase, MG algorithm involves application of genetic operators (reproduction, crossover and mutation) on every individual in the population. Being a mutually independent task, this can be done by spawning a new process for each individual. So, a parallelized version of MG algorithm on a multiprocessor machine can lead to drastic improvement.

References

[1] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[2] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI, USA: University of Michigan Press, 1975.

[3] R. S. Sutton and A. G. Barto, *An Introduction to Reinforcement Learning*. Cambridge, MA: MIT Press, 1997.

[4] R. E. Bellman, *Dynamic Programming*. Dover Publications, Incorporated, 2003.

[5] W. Zheng, J. Shu, X. Deng, and Y. Gu, "Parallel computing method of valuing for multi-asset european option," in *Proceedings of the 2003 International Conference on Computational Science: Part-II*, ser. ICCS'03. Berlin, Heidelberg: Springer-Verlag, 2003, pp. 3–9. [Online]. Available: http://portal.acm.org/citation.cfm?id=1760988.1760990

[6] L. Davis, "Job shop scheduling with genetic algorithms," in *Proceedings of the 1st International Conference on Genetic Algorithms*. Hillsdale, NJ, USA: L. Erlbaum Associates Inc., 1985, pp. 136–140. [Online]. Available: http://portal.acm.org/citation.cfm?id=645511.657084

[7] D. Smith, "Bin packing with adaptive search," in *Proceedings of the 1st International Conference on Genetic Algorithms*. Hillsdale, NJ, USA: L. Erlbaum Associates Inc., 1985, pp. 202–207. [Online]. Available: http://portal.acm.org/citation.cfm?id=645511.657091

[8] D. E. Goldberg, "Dynamic system control using rule learning and genetic algorithms," in *Proceedings of the 9th International Joint Conference On Artificial Intelligence*. Morgan Kaufmann, 1985, pp. 588–592.

[9] K. A. De Jong, "An analysis of the behavior of a class of genetic adaptive systems." Ph.D. dissertation, Ann Arbor, MI, USA, 1975, aAI7609381.

[10] D. E. Goldberg and J. R. Lingle, "Alleles loci and the traveling salesman problem," in *Proceedings of the 1st International Conference on Genetic Algorithms*. Hillsdale, NJ, USA: L. Erlbaum Associates Inc., 1985, pp. 154–159. [Online]. Available: http://portal.acm.org/citation.cfm?id=645511.657095

[11] C. L. Karr, "Genetic algorithms for modelling, design, and process control," in *Proceedings of the second international conference on Information and knowledge management*, ser. CIKM '93. New York, NY, USA: ACM, 1993, pp. 233–238. [Online]. Available: http://doi.acm.org/10.1145/170088.170139

[12] "Discrete and combinatorial optimization: Universität heidelberg." [Online]. Available: comopt.ifi.uni-heidelberg.de

[13] J. Cirasella, D. S. Johnson, L. A. McGeoch, and W. Zhang, "The asymmetric traveling salesman problem: Algorithms, instance generators, and tests," in *Revised Papers from the Third International Workshop on Algorithm Engineering and Experimentation*, ser. ALENEX '01. London, UK: Springer-Verlag, 2001, pp. 32–59. [Online]. Available: http://portal.acm.org/citation.cfm?id=646679.702307

[14] M. F. Tasgetiren, P. N. Suganthan, and Q.-Q. Pan, "A discrete particle swarm optimization algorithm for the generalized traveling salesman problem," in *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, ser. GECCO '07. New York, NY, USA: ACM, 2007, pp. 158–167. [Online]. Available: http://doi.acm.org/10.1145/1276958.1276980

[15] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, no. 1-2, pp. 99–134, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/B6TYF-3TT0072-4/2/25a6eb0d0c45a93c7533a565cdd2595b

[16] D. Whitley, V. S. Gordon, and K. Mathias, "Lamarckian evolution, the baldwin effect and function optimization." Springer-Verlag, 1994, pp. 6–15.

[17] L. Idoumghar and R. Schott, "A new hybrid ga-mdp algorithm for the frequency assignment problem." in *ICTAI'06*, 2006, pp. 18–25.

SESSION

ARTIFICIAL INTELLIGENCE AND COGNITIVE SCIENCE

Chair(s)

TBA

Metacognition and Metamemory Concepts for AI Systems

Dr. James A. Crowder, Shelli Friess, MA, NCC

Raytheon Intelligence and Information Systems
16800 E. Centretech Parkway, Aurora, Colorado 80011

Abstract - *Recent advances in cognitive computing suggest that computers can do as well as humans, even across multiple information sources and information types [4]. This work establishes an artificial cognitive neural framework to provide Metacognitive and Metamemory processing concepts similar to how the human brain operates to process, categorize and link information. Metacognition provides an Artificial Intelligence system with a sense of Self-Analysis, or Introspection, allowing the system to "think about what it thinks." Metamemory is a concept of an Artificial Intelligence system's memory capabilities and strategies that can aid in memory representation, retention, mining, retrieval, as well as the processes involved in memory Self-Monitoring. Metamemory constructs for an Artificially Intelligent system has important implications about how the system learns and uses memory. For example, the system can make a judgment on whether it has enough information to complete a mission, known as "judgments of learning." Presented here will be the derivation and application of Metacognition and Metamemory concepts to real-time system utilizing Artificial Intelligence and Intelligent Information Agents, including capture and utilization of Emergent Behavior, or concept derivation, allowing a system to preserve its own attention, avoiding interruption and distraction during the analytical process [1, 2].*

1 Introduction: Artificial Cognition

Metacognition, or ***Knowledge of Cognition***, refers to what a system knows about its own cognition or about cognition in general. In short, it describes the system's ability to think about how and what it thinks. It includes three different kinds of metacognitive awareness: declarative, procedural, and conditional knowledge [8, 9].

- **Declarative Knowledge**: refers to knowing "about" things,
- **Procedural Knowledge**: refers to knowing "how" to do things, and
- **Conditional Knowledge**: refers to knowing the "why" and "when" aspects of cognition.

We can classify Knowledge of Cognition into three components [2]:

- **Metacognitive Knowledge**: (also called Metacognitive awareness) is what the system knows about itself as a cognitive processor [13].
- **Metacognitive Regulation**: is the regulation of cognition and learning experiences through a set of activities that help the system control its learning [14]. This may be based on its understanding of its own "knowledge gaps."
- **Metacognitive Experiences**: are those experiences that have something to do with the current, on-going cognitive endeavors (current mission).

As in humans, we assume that direct inferences are necessary for cognitive processing and Metacognition [12]. For a normal, healthy system, if a cognitive function, ***F***, activates a functional cognitive area ***A***, it always activates ***A***. We utilize this phenomenon to create an initial Conceptual Ontological Framework (Figure 1) of cognitive instances for AI systems.

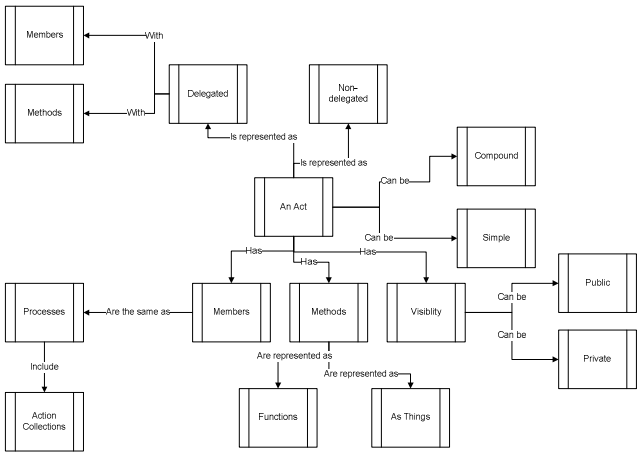


Figure 1 – Conceptual Cognitive Instance Framework

Each cognitive instance contains expectation derived from experiences the system has had [17]. We can deconstruct the cognitive instance expectations, based on

the concepts and skills that drive the expectations, as depicted in Figure 2.

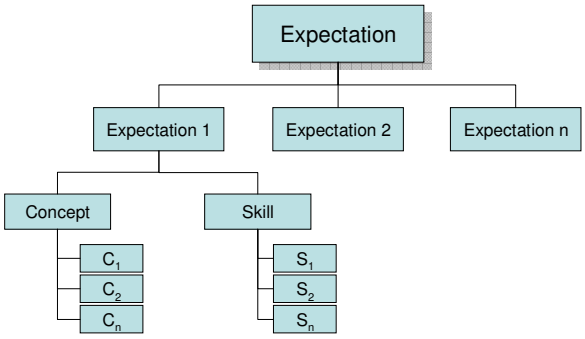


Figure 2 – The ACNF Cognitive Lower Ontology

The nature of Metacognition is to understand the structure of one's own cognitive processes, i.e., the ability to think about what we think. To an AI system, the beginnings of Metacognition are founded in understanding that each action the system might take is broken into sub-actions, each of which may have a separate and different contexts (see Figure 3).

Here we define the structures and methodologies required to provide Metacognition and Metamemory capabilities to AI systems [7]. Such capabilities will revolutionize autonomous systems, providing the capability for ***Self-Assessment*** and ***Self-Diagnosis***.

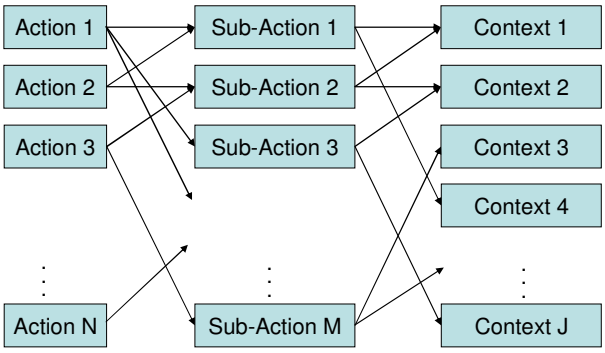


Figure 3 – The Structure of the Cognitive Action Process

2 Metacognition: Cognitive Self-Awareness and Assessment

In order to create cognitive self-awareness and assessment in AI systems, a formal cognitive neural framework is required for determining levels of cognitive granularity and to formalize methodologies for assessing the closeness of cognitive relationships within the AI system. This is facilitated through an Artificial Cognitive Neural Framework, which is a hybrid, fuzzy-neural processing system using genetic learning algorithms. This processing system uses a modular artificial neural architecture, depicted in Figure 4 [6]. This architecture is based on a mixture of neural structures that add flexibility and diversity to overall system capabilities. In order to provide an artificially intelligent processing environment that is continually adaptable, we believe the system must possess the notion of artificial emotions that allow the processing environment to "react" in real-time as the systems outside the environment change and evolve recursively as recombinant knowledge assimilation [18].

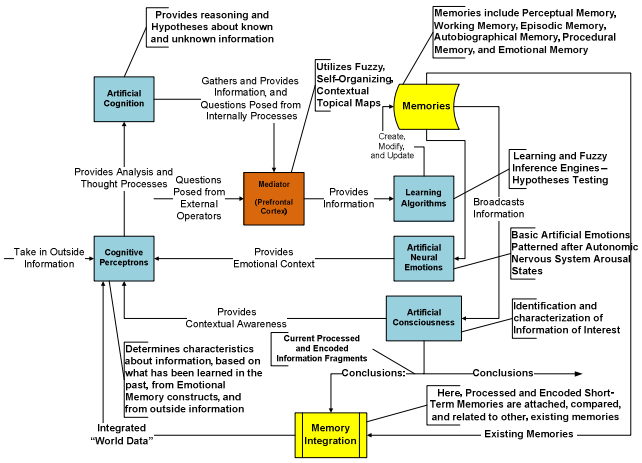


Figure 4– The Artificial Cognitive Neural Framework

In the ACNF, first the unconscious artificial neural perceptrons, each working toward a common goal, form a coalition. These coalitions will vie for access to the problem to be solved. The resources available to these coalitions depend on their combined nervous system state, which provides information on the criticality of their problem to be resolved. This nervous system state is defined by the Autonomic Nervous System States chart [18]. The purpose of the ACNF is to:

- Provide an architectural framework for "conscious" intelligent software agents [3].
- Provide a "plug-in" framework for the domain-independent portions of the Metacognitive mechanisms.
- Provide an easily customizable framework for domain-specific portions of the Metacognitive mechanisms.
- Provide the cognitive mechanisms for behaviors and artificial emotions for the conscious intelligent software agents [15, 16].

These constructs provide the necessary components of an AI system that has the capability of Metacognitive processes. As explained earlier, this allows the AI system to possess cognitive self-awareness and self-assessment processes. Figure 5 illustrates the information flow through the ACNF Metacognitive processes.

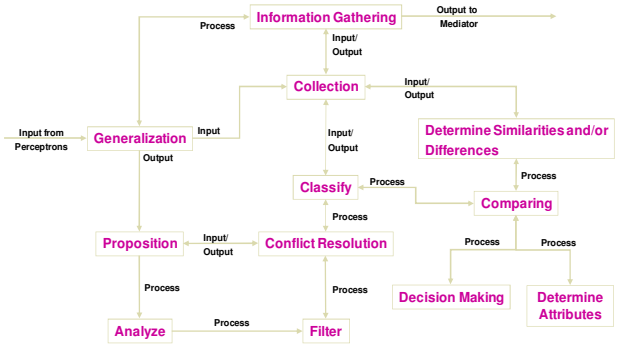


Figure 5 – Cognitive Self-Awareness and Self-Assessment

3 Metamemory: Cognitive Understanding of Learning

Metamemory: is the concept of an AI system's memory capabilities and strategies that can aid in memory representation, retention, mining, and retrieval, as well as the processes involved in memory self-monitoring [14]. A system's self-awareness of memory has important implications about how the system learns and uses memory [11, 12]. For example, the system can make a judgment on whether it has enough information to complete a mission, known as "judgments of learning."

Metamemory concepts for AI system resemble cognitive map constructs that utilize registries for hybrid topical map assimilation [13]. The Metamemory registry allows Cognitive Creation and Discovery within the overall ACNF and drives the conscious analytic inference engines of the system. Figure 6 illustrates these Cognitive Creation and Discovery Actions.

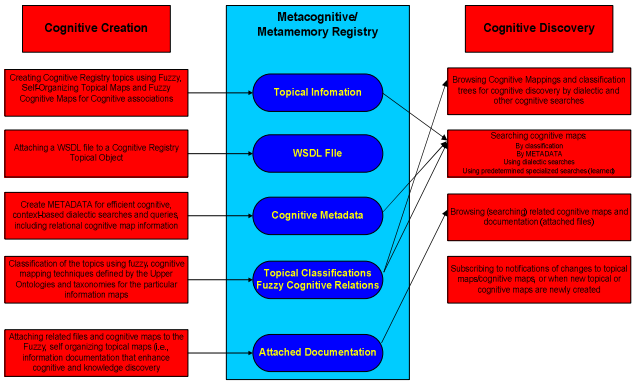


Figure 6 – Cognitive Creation and Discovery Actions

The Metamemory Registry has the following properties:

- It has a repository that contains all types of cognitive maps.
- It has a registry that contains metadata describing cognitive topics and cognitive maps, much like the library's card catalog contains information describing the published content on its book shelves.
- The Cognitive registry-repositories work together to offer a unified prefrontal cortex, much like multiple libraries can participate in a cooperative network and offer a unified service.

This Artificial Prefrontal Cortex registry-repository provides governance capabilities that enable definition and enforcement of cognitive policies governing the content and usage of the cognitive and topical maps by the Intelligent Software Agent framework across the AI enterprise. The cognitive registry-repository contains all Metacognitive and Metamemory constructs for cognitive components available within the AI, multi-agent framework.

The registry-repository framework provides discovery capabilities that are extensible and can accommodate the simplest to the most complex domain-specific cognitive discovery queries among the Intelligent Software Agents.

- Specifically, its discovery queries do not need to be predefined.
- Instead, it provides an ad hoc query syntax supporting complex predicates that can be combined using possibilistic logic.

As more and more cognitive components are reused by Intelligent Software Agents, the task of tracking the network of dependencies between cognitive topics becomes more challenging and significant. This is another challenge that is made easier by an Metacognitive and Metamemory registry-repository where inter-topical dependency information can easily be managed as relationships between cognitive maps. A cognitive registry-repository provides a set of standard relationship types, but also allows the definition of additional relationship types based on specific requirements within the overall Metacognitive framework.

The nature of these Metacognitive and Metamemory frameworks is that they evolve over time as the system learns and evolves. Cognitive components evolve over time for a variety of reasons, as information is learned and the system evolves. A cognitive component's evolution may involve changes in its topical references and/or interfaces. Changes to a cognitive interface need much more careful management because of the potential impact to existing Intelligent Software Agents. These changes will require a new version of the cognitive maps to be deployed, while maintaining the older cognitive until its Intelligent Software Agents have had time to migrate to the new version according to their own needs and schedule,

based on current analytical of mission drivers. New versions of a cognitive or topical map or a cognitive component also typically require publication of corresponding new versions of its cognitive information artifacts, much in the way new memories and thoughts must be categorized, catalogues, and correlated within the human brain.

The Metacognitive and Metamemory registry-repository provides a cognitive change notification capability that allows interested Intelligent Software Agents to create subscriptions to events within the registry-repository that may be of interest to them. Such a capability allows an Intelligent Software Agent to be flexible enough to express precisely the types of events that are of interest to the Agent.

Cognitive Visibility and Governance: the AI Metamemory system must provide cognitive visibility and governance across the entire ACNF framework allowing the AI system to create and process Metacognitive and Metamemory component models and store them. This enables cognitive governance within the ACNF Multi-Intelligent Software Agent framework. This allows transparency into the system's cognitive semantics and allows conceptualization across the system of cognitive metadata. These cognitive registry-repository systems allow easy access to cognitive topics and maps across the system, and facilitate knowledge gap analysis. Major functions of the overall Metamemory and Metacognitive systems are:

Cognitive Provisioning: the ACNF framework provides Cognitive Intelligent Software Agent interfaces and Metacognitive and Metamemory Metadata. This allows seamless Cognitive Integration across the AI system. The ACNF mediator allows Cognitive Metadata to be utilized in one central location. This enables the reuse of Cognitive and Memory artifacts and provides end-to-end Intelligent Software Agent support. It allow provides governance of Metacognitive and Metamemory definitions within the system.

Cognitive Process Integration: The Metacognitive and Metamemory information defined in the cognitive registry-repository is utilized to allow the system to create "what if" cognitive scenarios to facilitate self-assessment. These scenarios allow the system to correlate information and allow cognitive integration in a heterogeneous knowledge environment. This also promotes cognitive knowledge collaboration among the Intelligent Software Agents. All of this provides support to allow the system to define the cognitive integration scenarios.

Cognitive Composition: Cognitive Composition tools are provided to allow the system to compose queries of registry-repository information to discover new cognitive and memory components. This facilitates cognitive composite development within the overall AI cognitive framework. These tools reduce risk in cognitive discovery and allow easy knowledge discovery, consumption, and composition.

Figure 7 provides a Metacognitive and Metamemory Cognitive registry-repository Use Case, illustrating the various types of Intelligent Software Agents utilized within the ACNF. This Use Case describes the functions each type of Intelligent Software Agent contributes to the overall Metacognitive and Metamemory information flow within the AI system.

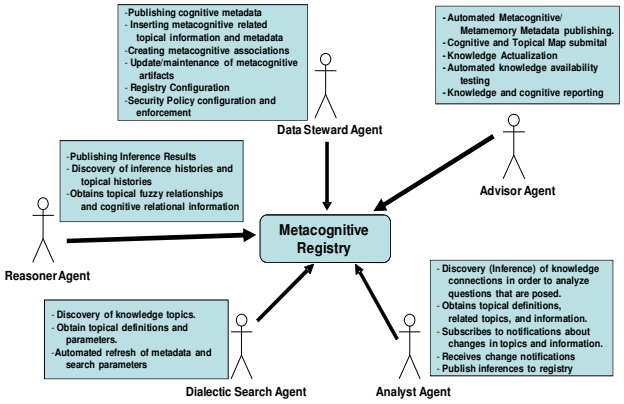


Figure 7 – Metacognitive and Metamemory Cognitive Registry Use Case

4 Metacognitive and Metamemory Argument Structures

Metacognitive and Metamemory Self-Awareness and Self-Assessment rely on the system's ability to pose "what-if" cognitive scenarios within its cognitive system. This requires argument structures that allow the AI system to pose hypotheses and determine the validity of these hypotheses. For this ability, we chose a Dialectic Argument Structure. The Dialectic Argument Search uses a Toulmin Argument Structure to find related information that develops a larger argument, or cognitive lead. The Dialectic Argument Search (DAS) has four components:

- **Information**: in support of the hypothesis or rebutting the hypothesis.
- **Warrant and Backing**: explaining and validating the hypothesis.
- **Claim**: defining the hypothesis itself.
- **Fuzzy Inferences**: relating the information to the hypothesis.

The DAS serves two distinctive purposes:

- First, it provides an effective basis for mimicking human reasoning.

- Second, it provides a means to glean relevant information from the Cognitive Topical Maps and transform the information into actionable intelligence, or practical cognitive knowledge.

These two purposes work together to provide a cognitive system that captures the capabilities to analyze and sort diverse information and cognitive clues that drive Emotional Memory and the Metamemory process in general [5, 10].

As cognitive knowledge are gathered, a cognitive lattice develops and an aggregate possibility is computed for/against the hypothesis using output from the fuzzy inference engines to provide fuzzy membership values of the support and rebuttal of the hypothesis [4]. This computation is based on Renyi's entropy theory, utilizing joint information membership metrics to generate a robust measure of hypothesis "possibility." Figure 8 illustrates the DAS structure.

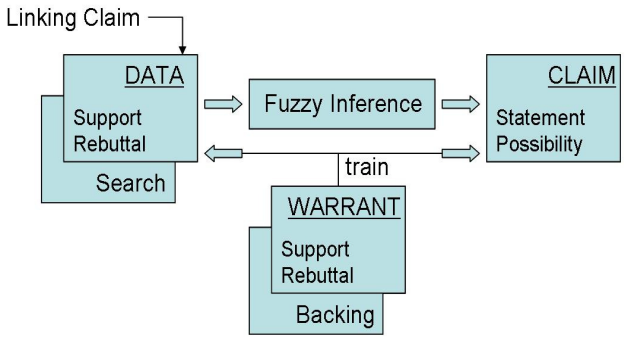


Figure 8 – The Dialectic Argument Search Structure

Intelligent Software Agents for the DAS, called Search Agents, learn fuzzy patterns and use these to evaluate cognitive information found by the DAS. Any information that does not quite fit is directed to a "cognitive sandbox" where peer agents can exercise a more rigorous adaptation routine to search for alternative hypotheses. The principle requirements addressed by the use of the DAS agents are:

- Ability to learn and adapt to changes in the AI system's surrounding environment.
- Capture the cognitive knowledge within the Metacognitive and Metamemory structures for reuse.
- Sharing of information and learning between Intelligent Software Agent peers.
- Hypothesize with "what-if" cognitive scenarios.
- Remember so as to avoid old mistakes and false leads.

Figure 9 illustrates the entire DAS Software Agency within the ACNF that drives the Metacognitive and Metamemory structures within the AI system. The DAS utilizes information fragments gathered from the topical cognitive maps to "argue" for and against the hypothesis. The main processing structures within the DAS are:

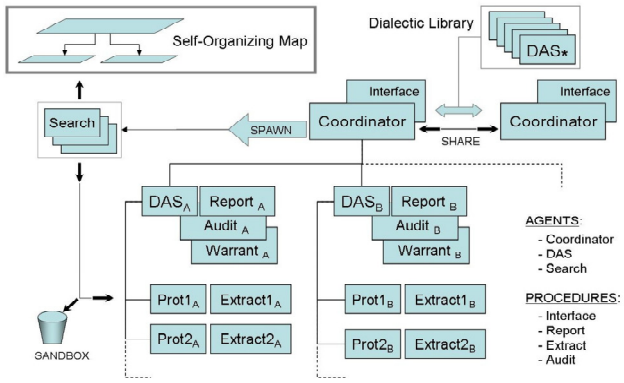


Figure 9 – The Intelligent DAS Software Agency

- **Dialectic Proof**: the opinion of individual experts, adhered to with variable intensity.
- **Toulmin Argument Structure**: non-analytic, plausible, practical reasoning.
- **Situational Reasoning**: find and model hypotheses using argument-chains.

The Intelligent Software Agents form agent Hierarchies for each cognitive hypothesis (see Figure 10). The agents are part of an ecosystem that are constantly having to adapt to the changing information and cognitive knowledge environment. The Intelligent Software Agent attributes that drive the DAS are:

- Agents search for information over an irregular topology.
- Agents compete and collaborate to increase their value; successes and failures are remembered.
- Agents evolve or die and exhibit different levels of consciousness.

Genetic Algorithms are used to evolve agents, maximizing the effectiveness of the cognitive ecosystem. In order to facilitate intelligent transmittal of learned emotions and emotional context, Emotional Markup Language (EML) is utilized within the system for transmittal of emotional information, creating a cognitive social intelligence, as depicted in Figure 11.

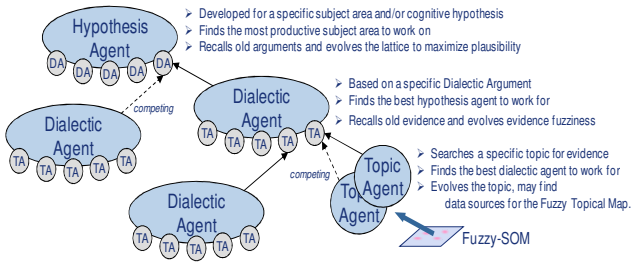


Figure 10 – The DAS Agent Hierarchy

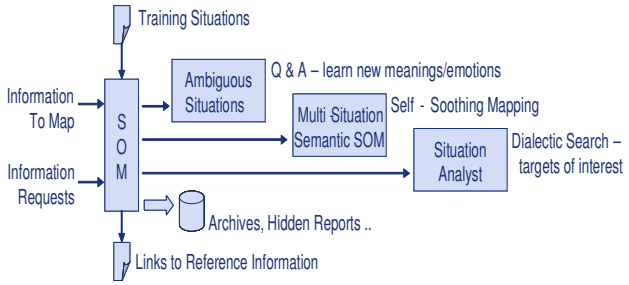


Figure 11 – Artificial Social Intelligence

The combination of these structures provides the overall framework to provide a collective Social Intelligence within the Intelligent Software Agent architecture. Figure 12 illustrates the information flow between the Intelligent Software Agents.

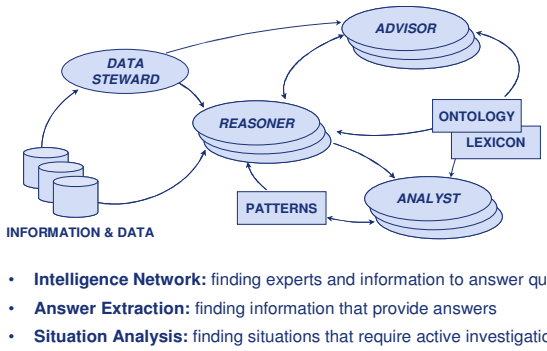


Figure 12 – Inter-Agent Information Flow

5 Conclusions and Discussion

The Metacognitive and Metamemory concepts presented here constitute the beginnings of a framework to allow real autonomous AI systems that can think, reason, adapt, evolve, and provide self-awareness and self-assessment within the AI cognitive framework. The artificial cognitive neural framework described provides analysis, reasoning and reporting capabilities (cognitive intelligence). Providing Metacognitive and Metamemory structures and processes within an autonomous AI system have the potential, I believe, to revolutionize AI and will allow full autonomy to be achieved.

6 References

1. Crowder, J. A., "*Adaptive Learning Algorithms for Functional Constraints on an Evolving Neural Network System*." NSA Technical Paper CON_0013_2002_003 (2002).
2. Crowder, J. A., "*Machine Learning: Intuition (Concept) Learning in Hybrid Genetic/Fuzzy/Neural Systems*." NSA Technical Paper CON_0013_2003_009 (2003).
3. Crowder, J. A., "Operative Information Software Agents (OISA) for Intelligence Processing." AIAA Aerospace@Infotech 2010 Conference.
4. Crowder, J.A., Friess, S., "Artificial Neural Diagnostics and Prognostics: Self-Soothing in Cognitive Systems." International Conference on Artificial Intelligence, ICAI'10 (July 2010).
5. Crowder, J. A., Friess, S., "Artificial Neural Emotions and Emotional Memory." International Conference on Artificial Intelligence, ICAI'10 (July 2010).
6. Crowder, J. A., "Flexible Object Architectures for Hybrid Neural Processing Systems." International Conference on Artificial Intelligence, ICAI'10 (July 2010).
7. Crowder, J. A., Carbone, J, "The Great Migration: Information to Knowledge using Cognition-Based Frameworks." Springer Science, New York (2011)
8. Marsella, S., and Gratch J., "*A Step Towards Irrationality: Using Emotion to Change Belief*." 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems, Bologna, Italy (July 2002).
9. Newell, A., "*Unified Theories of Cognition*." Cambridge MA: Harvard University Press (2003).
10. Damasio A (1994) Descarte's error: Emotion, reason, and the human brain. New York: Gosset/Putnam.
11. Davis M, Whalen PJ (2001) The amygdala: vigilance and emotion. Mol Psychiatry 6:13-34.
12. Eichenbaum H (2002) The cognitive neuroscience of memory. New York: Oxford University Press.
13. Kosko, G., "*Fuzzy Cognitive Maps*," International Journal of Man-Machine Studies, 24: 65-75.
14. LaBar KS and Cabeza (2006) Cognitive neuroscience of emotional memory. Nat Rev Neurosci 7: 54-64.
15. LeDoux JE (1996) The Emotional Brain. New York: Simon and Schuster.
16. LeDoux JE (2000) Emotion circuits in the brain. Annu Rev Neurosci 23:155-184.
17. LeDoux JE (2002) Synaptic Self: How our brains become who we are. New York: Viking.
18. Levine, P., "Walking the Tiger: Healing Trauma." North Atlantic Books, Berkeley, CA (1997).

The Artificial Prefrontal Cortex: Artificial Consciousness

Dr. James A. Crowder, Shelli Friess, MA, NCC

Raytheon Intelligence and Information Systems
16800 E. Centretech Parkway, Aurora, Colorado 80011

Abstract - *The purpose of this paper is to describe a new neural framework for Artificially Intelligent systems, the Artificial Cognitive Neural Framework (ACNF), which allows for "conscious" software agents within the AI processing environment. "Conscious" software agents are autonomous agents that range in functionality and are situated in the processing environment. They sense the environment and act on it over time, in pursuit of their own agenda, based on their evolving constraints. As they evolve it is possible for them to change what they sense at a later time. These "conscious" agents are also "cognitive" agents, in that they are equipped with constructs for concept formation, consciousness, basic emotions, and short & long-term memories [14]. The long-term memories provide identification, recognition and categorization functions, as well as identification of feelings [21]. The short-term memories provide preconscious buffers as a workspace for internal activities. A transient episodic memory is also provided as a content-addressable associative memory with a moderately fast decay rate.*

This provides the architectural framework for an Artificial Prefrontal Cortex (APC), which provides cognitive intelligence for AI processing systems and allows for rapid analysis, reasoning, and reporting capabilities. The APC facilitates information, intelligence, and memory integration and allows faster accommodation and delivery of knowledge and knowledge characteristics across the system [16].

1 Introduction

The prefrontal cortex has long been suspected to play an important role in cognitive control, in the ability to orchestrate thought and action in accordance with internal goals. Cognitive control stems from the active maintenance of patterns of activity in the prefrontal cortex that represent goals and the means to achieve them [19]. They provide bias signals to other cognitive structures whose net effect is to guide the flow of activity along neural pathways that establish the proper mappings between inputs, internal states, and outputs needed to perform a given task. The Prefrontal Cortex is integral to planning complex cognitive behaviors, personality expression, decision making and moderating correct social behavior [21]. The basic activity of this brain region is considered to be orchestration of thoughts and actions in accordance with internal goals [10].

Here we present structure within the architecture to provide an APC and provides the structure and context for artificial feelings and emotions within the overall AI system [17, 18]. Here we discuss the roles they can play in an intelligent software agent performing a practical, real world task. These agents would be actively involved in every instance of action selection, and at least potentially involved in each learning event [11]. The pervasive, central role that feelings and emotions would play in the control structure of these conscious software agents mimics the roles they play in human cognition, and, over time, may give rise to clarifying hypotheses about human decision-making and several forms of human learning [12, 13].

The functions carried out by the prefrontal cortex area can be described as executive functions. Executive functions relate to abilities to differentiate among conflicting thoughts, determine good and bad behavior, better and best, same and different, future consequences of current activities, working toward a defined goal, prediction of outcomes, expectation based on actions, and social "control" [8]. The prefrontal cortex is of significant importance when top-down processing is needed. Top-down processing by definition is when behavior is guided by internal states or intentions. All of these are driving toward the cognitive concept of "mindfulness:"

- Mindfulness: an awareness that lets us see things as they truly are without distortion or judgment, giving the most insightful explanation of how mindfulness can change not only our lives, but the very structure of our brains.

In order for AI systems to be truly autonomous, we must give them these "executive functions" abilities. One of the cognitive concepts that will be necessary for a truly autonomous system is the ability to top-down processing. To take an understanding of the mission or task at hand, from this define goals and prediction of outcomes, and utilize this knowledge to define the system behaviors needed to meet the mission or task goals. For an autonomous AI system, executive management and strategic knowledge would be essential. The executive management autonomous system processes would involve planning, monitoring, evaluating and revising the system's own cognitive processes and products. Strategic knowledge would involve knowing what tasks or operations to perform (factual or declarative knowledge), knowing when and why to perform the tasks or operations (conditional or contextual knowledge) and knowing how to perform them (procedural or methodological knowledge). Both executive management and strategic knowledge capabilities are required for the system to autonomously self-regulate its own thinking and learning [2].

Here we propose a model for an Artificial Prefrontal Cortex as part of an overall Artificial Cognitive Neural Framework (ACNF) required for real AI autonomous system control. We will begin with a discussion of the ACNF and then introduce a Hidden Markov Model of an Artificial Prefrontal Cortex, utilizing fuzzy possibilistic logic to drive the system between cognitive states.

2 The Artificial Cognitive Neural Framework

As knowledge and cognitive context increases within the AI system, a formal framework for dealing with increasing and decreasing levels of cognitive granularity is necessary to learn, understand, and store the closeness of cognitive relationships [7]. We deal with these abilities utilizing a hybrid, fuzzy-neural processing system with genetic learning algorithms [1]. This processing system uses a modular artificial neural architecture. This architecture is based on a mixture of neural structures with Intelligent Software Agents that add flexibility and diversity to the overall system capabilities. In order to provide an artificially intelligent processing environment, we believe the system should possess the notion of artificial emotions that allow the processing environment to "react" in real-time as the systems environment changes and evolves. This hybrid fuzzy-neural processing framework is called the Artificial Cognitive Neural Framework (ACNF) [5]. Figure 1 illustrates the ACNF which has analogies to an AI blackboard system, except that it is greatly extended to allow for system-wide action selection.

The three main subsystems within the architecture are the Mediator, the Memory System, and the Cognitive System [2]. The Mediator gathers information and facilitates communication between agents. Hence, each cognitive decision is handled by the Mediator (the Artificial Prefrontal Cortex) which takes information from perceptrons and from coalitions of perceptrons and updates the short-term, long-term and episodic memories or pedigree [9]. The information available in memory (what the system has learned) is continually broadcast to the conscious perceptrons that form the cognitive center of the system (i.e., they are responsible for the cognitive functionality of perception, consciousness, emotions, processing, etc.) [10].

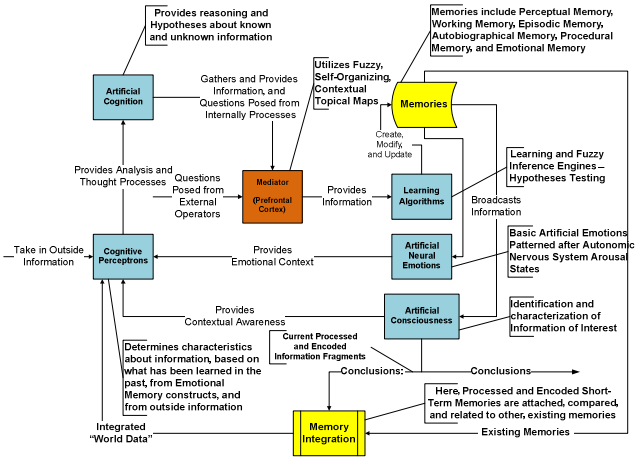


Figure 1– The Artificial Cognitive Neural Framework

The use of an ACNF for analysis, reasoning, and reporting provides the "Cognitive Intelligence" to allow the top-down executive processing required for real-time autonomous operations with AI systems [7]. The ACNF utilizes Fuzzy-Self Organizing Contextual Topical Maps that allow information fragments to be put together to form memories based on topical information. The Mediator, or Artificial Prefrontal Cortex, has the following properties:

- Facilitates Information, Intelligence, and Memory Integration.
- Allows faster accommodation and delivery of cognitive knowledge and knowledge characteristics.
- Increases cognitive flexibility, allowing rapid adaptation to changing environments.
- Allows scalability of the overall AI system.
- Reliably provides cognitive information across domains, even after a software, network, or hardware failure.
- Provides a topical, information, cognitive knowledge hosting and management infrastructure that is highly distributed, yet manageable.

The Artificial Prefrontal Cortex is implemented or instantiated with Intelligent Information Software Agents. The logical flow is illustrated in Figure 2.

3 The Artificial Prefrontal Cortex

The Artificial Prefrontal Cortex provides governance capabilities that enable definition and enforcement of cognitive policies governing the content and usage of the cognitive and topical maps by the Intelligent Software Agent framework across the AI enterprise.

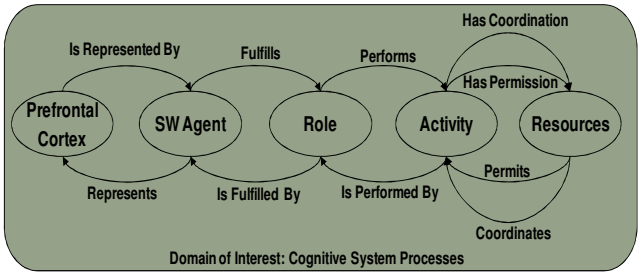


Figure 2– Artificial Prefrontal Cortex Inference Flow

In order to understand the cognitive interactions that must occur within an Artificial Prefrontal Cortex, a model was built to drive the Intelligent Software Agent framework that provides linkage between the major cognitive states within the cortex [4, 5]. Figure 3 illustrates this model. The cognitive processes represented are based on AI interpretations of Dr. Peter Levine's Autonomic Nervous System States [20].

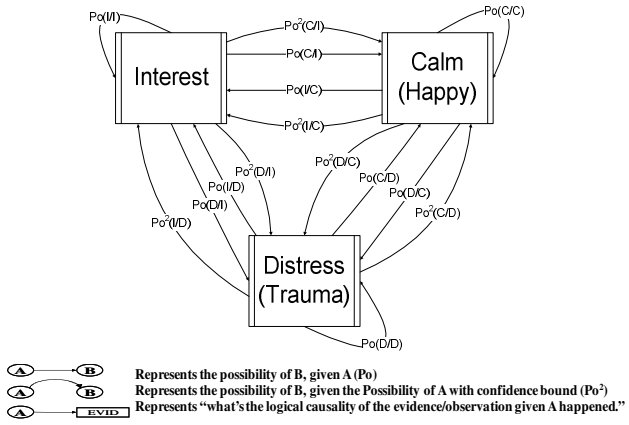


Figure 3– Artificial Prefrontal Cortex Affected State Model

Detecting cognitive process information within the ACNF begins with sensors that capture information about the system's physical or cognitive state or behavior. The information is gathered and interpreted by the cognitive perceptrons similar to how humans utilize cues to perceive cognitive states or emotions in others. The Artificial Prefrontal Cortex (APC) provides the possibilistic inferences for the system to transfer between cognitive states. The APC shown in Figure 3 illustrates only three cognitive states for display purposes only, but it is straight forward to include more states. The idea is that the APC makes it possible to transition between cognitive states at any instant, and transition between these states with certain possibilistics. These possibilistic parameters evolve over time, driven by the learning algorithms and how they affect both normal and emotional memories. These cognitive state transition conditional possibilistics provide the APC with the ability to make executive-level plans and move between cognitive states, each of which has its own set of priorities, goals, and motivations. The driving requirement for the APC is to create a truly autonomous AI system that can be used in a variety of applications like UAVs, intelligence processing systems, cyber monitoring and security systems, etc. In order to accomplish these tasks, the APC must have the following capabilities processes and acted on by the human prefrontal cortex:

Cue Familiarity: cue familiarity is the ability of the system to evaluate its ability to answer a question *before* trying to answer it [21]. In cue familiarity, the question (cue) and not the actual memory (target) becomes crucial for making cognitive judgments. This implies that judgments regarding cognitive processing and decisions would be based on an the system's level of familiarity with the information provided in the cue. This executive-level, top-down cognitive judgment requires APC abilities to allow the AI system to judge whether they know the answer to a question, i.e., is the system familiar with the topic or mission, allowing the system to to judge that they do not know the answer to a question which presents new or unfamiliar terms or conditions.

Cognitive Accessibility: cognitive accessibility suggests that the system's memory will be more accurate when the ease of cognitive processing (accessibility) is correlated with memory behavior (emotional memory). This implies that the quality of information retrieval depends on the system's density of knowledge on the topic or subject (or individual elements of informational content about a topic), since the individual elements of topical information differ in strength. The speed of access is tied to both density of knowledge and emotional memory responses to the information.

Cognitive Competition: cognitive competition can be described as three principles:

- The AI cognitive processing system (the brain) is activated by a variety of inputs (sensors). There is textual, audio, and visual (picture and video) information that compete for cognitive processing access.
- Competition occurs within the multiple cognitive processing subsystems and is integrated by the Intelligent Software Agents between the various cognitive processing subsystems.

- Competition can be assessed utilizing top-down neural priming within the APC, based on the relevant characteristics of the object at hand.

Cognitive Interaction: This combines cue familiarity and cognitive accessibility. In cognitive interaction, once cue familiarity fails to provide enough information to make cognitive inferences, cognitive accessibility in employed access extended memories and may employ stored emotional memory cues to access the required information to make the required cognitive inferences. This may result in slower response time that with cue familiarity alone.

4 Artificial Prefrontal Cortex Processing

In order to provide the APC with capabilities described above, processing constructs must be in place to allow cognitive inferences to be made, based on the information received, inferences and decisions learned, and an overall sense of priorities, goals, and needs. The following describes processing constructs that allow a viable APC to be constructed.

4.1 The Fuzzy, Self-Organizing, Contextual Topical Map

The Fuzzy, Self-Organizing, Contextual Topical Map (FSOCTM) is a general cognitive method for analyzing, visualizing, and providing inferences for complex, mulit-dimensional sensory information (textual, auditory, and visual). The FSOCTM is actually built on two, separate, Fuzzy, Self-Organizing topical Maps (FSOM). The first is a semantic FSOM that organizes the information semantically into categories, or topics, based on the derived eigenspaces of features within the information. Figure 4 illustrates an FSOM with information and topical "closeness" search hits designated. The larger hexagons denote topical sources that best fit the search criterion. The isograms denote how close the hits are to a particular cognitive information topic.

The FSOM information and topical closeness map has several important attributes:

- Image processing algorithms can be utilized to analyze the output of the FSOM
- Searches use contextual information to find cognitive links to relevant memories and information available.
- The FSOM is self-maintained and automatically locates input from relevant Intelligent Software Agents and operates unsupervised.

The high-level topical spaces are compared, within the APC to identifiable "eigenmoods" within the emotional memory. The resulting eigenspaces determine topics that are compared within the contextual FSOM to look for "closeness" of topics to be used in cognitive processing algorithms to determine the cognitive state that will be used to make inferences about the question or task being posed. The eigenspaces are estimated under a variety of emotional memory conditions and their dependencies on external inputs and cognitive factors determined. Eigen Trajectories are then characterized, capturing the dynamic aspects of relationships between topical closeness and the information and memories available.

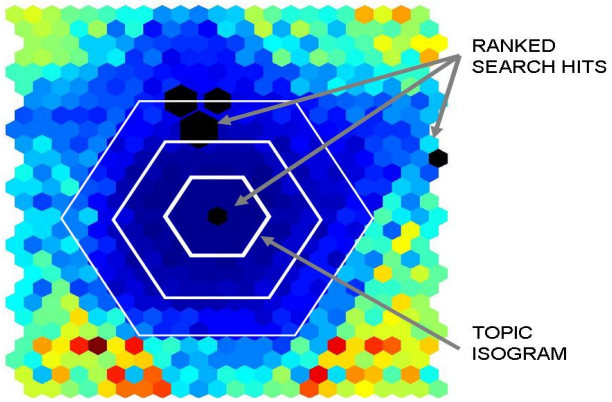


Figure 4– The Fuzzy, Self-Organizing Topical Map

The high-level topical spaces are compared, within the APC to identifiable "eigenmoods" within the emotional memory. The resulting eigenspaces determine topics that are compared within the contextual FSOM to look for "closeness" of topics to be used in cognitive processing algorithms to determine the cognitive state that will be used to make inferences about the question or task being posed. The eigenspaces are estimated under a variety of emotional memory conditions and their dependencies on external inputs and cognitive factors determined. Eigen Trajectories are then characterized, capturing the dynamic aspects of relationships between topical closeness and the information and memories available.

Once the FSOM is created, the resultant topical eigenspaces are mapped to the larger FSOCTM to show cognitive influences and ties to larger cognitive processes and other memory information, as depicted in Figure 5.

The value of superimposing the FSOCTM onto the SOM is that it defines it defines the cognitive information domain's ontology, and enables the use of a Topic Map Query Language (TMQL) within the APC. The Topic Map enables end APC to rapidly search information conceptually. It also enables sophisticated dialectic searches to be performed for them.

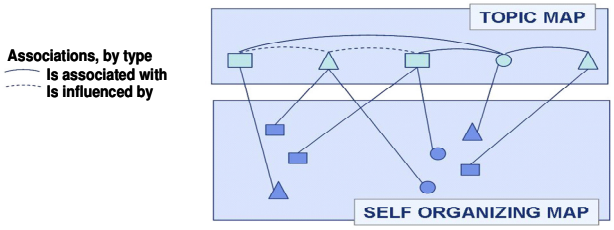


Figure 5– Superimposing the FSOCTM onto the FSOM

5 The Dialectic Search (DS)

The Dialectic Search uses the Toulmin Argument Structure to find and relate information and memories that develops a larger argument, cognitive inference. The Dialectic Search Argument (DSA), illustrated in Figure 6, has four components:

- **Information and Memories**: both in support of and rebutting the argument or hypothesis under analysis by the APC.
- **Warrant and Backing**: explaining and validating the hypotheis.
- **Claim**: defining the hypothesis itself
- **Fuzzy Inference**: relating the information/memories to the hypothesis.

The Dialectic Search serves two purposes:

- First, it provides an effective basis for mimicking human reason.
- Second, it provides a means to glean relevant information from the Topic Map and transform it into actionable cognitive intelligence.

These two purposes work together to provide an intelligent system that captures the capability of the human reasoning to sort through diverse information and find clues (based on Cue Familiarity discussed above). This approach is considered dialectic in that it does not depend on deductive or inductive logic, though these may be included as part of the warrant. Instead, the Dialectic Search depends on non-analytic inferences to find new possibilities based upon warrant examples. The Dialectic is dialectic because its reasoning is based upon what is plausible; the Dialectic Search is a hypothesis fabricated from bits of information fragments (memories) put together utilizing the topical maps and eigenspaces [3].

Once the available information has been assimilated by the Dialectic Search, information that fits the support and rebuttal requirements is used to instantiate a new claim or hypothesis. This claim is then used to invoke one or more new Dialectic Searches. The developing lattice forms the reasoning that renders the cognitive intelligence lead plausible and enables the possibility to be measured and cognitive inferences made within the APC.

The approach to cognitive intelligence inferencing within the APC is threefold:

- First the FSOCTM is investigated to semantically organize the diverse information collected and retrieved from memory.
- The map produced by the FSOM is utilized to enhance the APCs comprehension about the situations under analysis.
- Third, as the APC traverses the map to find related and relevant events, the results are used to create cognitive clues that are stored in the emotional memory for use under similar circumstances.

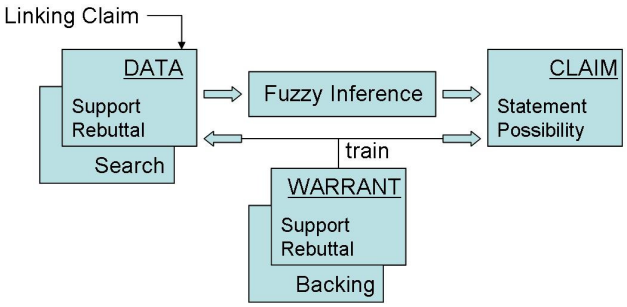


Figure 6– The Dialectic Search Structure

This approach mimics human intelligence, learning from Intelligent Software Agents using knowledge ontology to define particular knowledge domains (topics), having experts (intelligent information software agents) to cartographically label the FSOM to capture the meaning of the integrated information thus capturing the knowledge of each cognitive inference. The APC processing environment has three processing levels, illustrated in Figure 7 [6].

- The first will identify patterns of behavior that have been seen (or behavior similar in a "fuzzy" relational way) before.
- The second is an expanded pattern recognition that involves pattern discovery algorithms that augment patterns that are similar to known patterns but need additional information to describe the pattern divergences.
- The third is a full up pattern discovery paradigm to make sense of information that has not been previously described (how does the system find things it didn't know it was looking for.

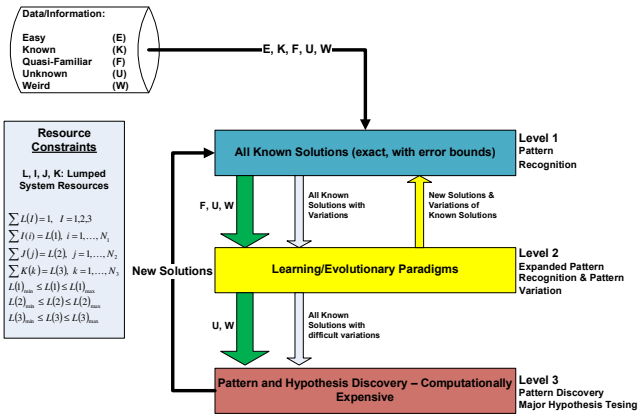


Figure 7– The APC Intelligent Software Agent Processing Levels

6 Conclusions and Discussion

The need to mimic human intelligence demands a polymorphic architecture that is capable of both hard and soft computing. The APC with the FSOCTM, soft computing, and utilizing the ACNF framework provides the structure that allows the APC to evolve and grows as it learns more about its environment. There is also a need to process streams of diverse information to provide terse vectors for FSOM and cognitive mapping [15]. This is accomplished through the use of the genetically evolving ACNF processing network.

Reason for using a FSOM approach is to ensure the results can be readily understood by the APC. The FSOM performs a critical role by collapsing multiple dimensions in information onto 2-dimensional space – a form that be more easily computed and understood by the APC, especially when it has been enhanced to include emotional memory information. As more information is acquired, it is mapped into an already understood structure within the ACNFs structure [5]. We believe the work outlined in this paper creates the foundation for further research into the creation of an Artificial Prefrontal Cortex, which we feel is essential to an actual learning, thinking, reasoning, and autonomous AI system.

7 References

1. Crowder, J. A., "Adaptive Learning Algorithms for Functional Constraints on an Evolving Neural Network System." NSA Technical Paper CON_0013_2002_003 (2002).
2. Crowder, J. A., "Machine Learning: Intuition (Concept) Learning in Hybrid Genetic/Fuzzy/Neural Systems." NSA Technical Paper CON_0013_2003_009 (2003).
3. Crowder, J. A., "Operative Information Software Agents (OISA) for Intelligence Processing." AIAA Aerospace@Infotech 2010 Conference.
4. Crowder, J.A., Friess, S., "Artificial Neural Diagnostics and Prognostics: Self-Soothing in Cognitive Systems." International Conference on Artificial Intelligence, ICAI'10 (July 2010).
5. Crowder, J. A., Friess, S., "Artificial Neural Emotions and Emotional Memory." International Conference on Artificial Intelligence, ICAI'10 (July 2010).
6. Crowder, J. A., "Flexible Object Architectures for Hybrid Neural Processing Systems." International Conference on Artificial Intelligence, ICAI'10 (July 2010).
7. Crowder, J. A., Carbone, J, "The Great Migration: Information to Knowledge using Cognition-Based Frameworks." Springer Science, New York (2011)
8. DeYoung, C. G., Hirsh, J. B., Shane, M. S., Papademetris, X., Rajeevan, N., and Gray, J. R. (2010). Testing predictions from personality neuroscience. Psychological Science, 21(6):820-828.
9. Marsella, S., and Gratch J., "A Step Towards Irrationality: Using Emotion to Change Belief." 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems, Bologna, Italy (July 2002).
10. Miller EK, Freedman DJ, Wallis JD (August 2002). "The prefrontal cortex: categories, concepts and cognition". Philos. Trans. R. Soc. Lond., B, Biol. Sci. **357** (1424): 1123–36.
11. Newell, A., "Unified Theories of Cognition." Cambridge MA: Harvard University Press (2003).
12. Damasio A (1994) Descarte's error: Emotion, reason, and the human brain. New York: Gosset/Putnam.
13. Davis M, Whalen PJ (2001) The amygdala: vigilance and emotion. Mol Psychiatry 6:13-34.
14. Eichenbaum H (2002) The cognitive neuroscience of memory. New York: Oxford University Press.
15. Kosko, G., "Fuzzy Cognitive Maps," International Journal of Man-Machine Studies, 24: 65-75.
16. LaBar KS and Cabeza (2006) Cognitive neuroscience of emotional memory. Nat Rev Neurosci 7: 54-64.
17. LeDoux JE (1996) The Emotional Brain. New York: Simon and Schuster.
18. LeDoux JE (2000) Emotion circuits in the brain. Annu Rev Neurosci 23:155-184.
19. LeDoux JE (2002) Synaptic Self: How our brains become who we are. New York: Viking.
20. Levine, P., "Walking the Tiger: Healing Trauma." North Atlantic Books, Berkeley, CA (1997).
21. Yang Y, Raine A (November 2009). "Prefrontal structural and functional brain imaging findings in antisocial, violent, and psychopathic individuals: a meta-analysis". Psychiatry Res **174** (2): 81–8. doi:10.1016/j.pscychresns.2009.03.012. PMID 19833485.

Why is missing what we need

Vitor Manuel Dinis Pereira[1]
[1]Department of Philosophy, University of Lisbon, Lisbon, Portugal

Abstract - *Access consciousness of agents as Homer and COMA cannot be separated from a body (Total Turing Test), but this body cannot be any aggregate of matter: human beings looking at these bodies and confuse us, to process them as to other humans. The phenomenological properties of the bodies of these agents, the way they appear to us, being indistinguishable from the phenomenological properties of human bodies. Our brain processes androids as human for two seconds (there are studies showing that in 70% of participants this is the case). It is for this that we need a theory of subjectivity and a theory of the body. By analogy with the concept of sound (for example), we need a conceptual expansion of the concept of the body: one expansion that does not involve conflict with the essential nature of subjective experience of the body.*

Keywords: phenomenal consciousness, total Turing test, artificial intelligence, neuroscience, body, androids

1 Introduction

When we talk about consciousness we can be understood as wanting to answer the following three questions.

Descriptive Question: what is consciousness? What are its main characteristics? And in what ways can these be better discovered, described and modelled?

Explanatory Question: how does consciousness manifests itself? It is an aspect of primitive reality and if it isn't, how does consciousness arises or could arise (be caused) by entities or processes that are non-conscious?

Functional Question: why does consciousness exist? Does it probably have a function and, if affirmative, what is that function? Does it have causal effects and if so, what type of effects? Does it make a difference for the performance (the operation) of the systems in which it is present and if so, why and how?

The topics in which we can subdivide the first of these questions are the following: subjectivity, the infallibility and omniscient of ours own mental states, intentionality, rationality, freedom, self-knowledge (features that are mental in nature – which apparently are only applicable to body sensations such as pain, perceptions as in visual, auditory, olfactory, tactile experiences and propositional attitudes such as beliefs and desires).

Our essay is mainly about the descriptive question: Artificial Intelligence (AI) is a way of discovering, describing and modelling some of the main features of consciousness – specifically the cognitive ones. But it is also about two other questions. AI help Cognitive Science researchers in explaining how consciousness emerges or could emerge (be caused) by non-conscious entities and processes (explanatory question), or if it makes any difference for the performance (the operation) of the systems in which it is present and if so, why and how (functional question).

But let's go more depth in the AI and in the Cognitive Science fields.

2 AI and Cognitive Science

The AI field can be outlined in the following way: machine learning, reasoning, knowledge representation, restriction fulfilment, search, planning and scheduling, agents, robotics, philosophical foundations, natural language processing, perception and vision, cognitive modelling, knowledge and applications engineering. The main core consists of the first three items: machine learning, reasoning, and knowledge representation. Now let's go to Cognitive Science.

Considered the following items. Perception and action, memory, care and consciousness, the so-called nuclear knowledge, classification, lexicon and ontology, learning, language and representation, choice, rationality and decision, culture and social awareness. The field we can outline with them is Cognitive Science, with AI as a it's proper part.

Cybernetics, Computer Sciences, Language Sciences, Neurosciences, Brain Sciences, Psychology, Biology, Philosophy, Mathematics, Physics, Engineering Sciences, in a way all of these sciences contribute to the study of the aforementioned items, to the study of human cognition, but also to the study of subjectivity.

Considering the descriptive, explanatory and functional questions about consciousness, a central notion in AI is that of agent. The idea of an agent is an ongoing and autonomously operating entity in an environment in which there are other processes and agents. In the seventies it was a closed and isolated entity. Currently it is an open, social and organizational entity. We are interested in knowing how an intelligent agent is designed. Usual questions are the following: how does it perceive, rationalize, decide, learn, how does it perform independently in a mutual environment of problems (specific agents for certain intervention domains)? Hereinafter we are interested in multiplying those agents and ask how it works that an enormous variety of those agents can articulate coherently in a multi-agent system (interaction and organization). The combination of these questions (and their answers) can be designated by the term "Distributed Artificial Intelligence (DAI)".

3 Consciousness

With respect to consciousness, it can be classified in the following three ways [1].

3.1 Access consciousness

Access consciousness: we have access consciousness of something if we have its representation, it can be transmitted to each part of the brain and in this way it can be used in our reasoning and (rational) control of our actions. It is likely that this is the type of consciousness that can be implemented in a machine. But we have the problem of debating whether the machine "actually" experiences something or not (and in this case "actually" is not clearly defined).

3.2 Phenomenal consciousness

Phenomenal consciousness: ***x*** is in a phenomenal conscious state if ***x*** experiences something that characterizes that state. The criterion widely used to talk about phenomenal consciousness is that of "there is something it is like to be in that state". For example, if we are phenomenally conscious of a bright blue sky, then it is because we are experiencing something that makes that mental state a phenomenal conscious state. This experience is the key concept of phenomenal consciousness. Block identifies the following three differences between access consciousness and phenomenal consciousness [1].

3.2.1 Differences between access and phenomenal

Access consciousness is completely defined by a representation (such as a logical agent clause that represents a concept or a fact). Phenomenal consciousness can also have representational component, but what identifies it is the experience of ***x*** (an agent) so that if ***x*** were not in this phenomenal conscious state it would not have the experience that it de facto has.

Access consciousness characterizes a mental state as a conscious state because their relations with other modules (in other words, access consciousness uses a functional way of classifying mental states as a conscious states). Being aware is being capable of reasoning and acting, of being stimulated and responding to those stimuli.

Phenomenal consciousness identifies types of conscious states. For example, all the sensations of pain are phenomenal conscious states of the same type, pain. But if we consider each pain from the perspective of access consciousness, each pain is a different conscious state because it causes different reactions, memories and inferences. To better illustrate the difference between access and phenomenal consciousness, Block describes cases of the first without the second and vice-versa, of access without phenomenal (a) and phenomenal without access (b) [1]. Those cases are, for example, the following.

(a) An individual can have his visual cortex damaged (have suffered an injury in the V1 area), there are things in his field of vision that he cannot see, the so-called blind spots, and even so respond with elevated exactness to questions concerning the properties of those visual stimuli. This pathology, called blind-sight [2], exemplifies the case of access consciousness without phenomenal consciousness. Phenomenologically it is not aware of anything, but this does not preclude it from representing those stimuli. Are your representations that enable it to respond to such visual stimuli.

But we can still give another example. The one from the BDI (Belief-Desire-Intention) agent, which does not have experiences: he is aware of everything in front of him but does not experience any of it (a discussion related to this example is the experience of thought of the Chinese Room [3]). His is access, not phenomenal consciousness.

(b) One case of phenomenal consciousness without access consciousness is, for example, in which we experience one additional sound, because of being so used to living with it, we do not represent it. Our friend used to the silence of the countryside could find it strange how we are able to live in the noise of the city. The reason is that we are not access conscious even though we are phenomenal conscious of it.

3.3 Awareness

Awareness of oneself, self-awareness: is the state of something when there is an internal representation of oneself. For example, a chimpanzee or a baby (baby around 2 years, 2 and one-half years old) is capable of recognizing itself in the mirror but a dog is not. It is likely when a dog looks at the reflected image (of itself) in the mirror, it is conscious of the phenomenal but it interprets the representation to which it has conscious access as another dog.

4 The main difficulty

Coelho asserts the need of a theory of subjectivity and a theory of the body [4]. The difficulty of the subjectivity theory can be illustrated in the following way: we are not capable of having the sensations of a bat (we are not bats) [5]. And the difficult thing about the theory of the body, in the following way: "robotic" organs are not organs from natural selection, but our brain is an organ of natural selection [6].

The theory that we need is a theory free from mental and physical concepts with which we currently operate. We have the theories that we have because we have the traditional (Cartesian) categories that we have. For example, Searle's thesis is that consciousness is a biological characteristics of the human brain and the brain of some animals, which are caused by neurophysiologic and form part, as any other biological characteristic (for example, digestion), of the natural biological order [7]-[8]. However, there are philosophers for whom Searle's thesis is too dualist. Nagel is one of those philosophers. Relating the physiological with the mental as cause and effect, Nagel's line of reasoning is that this does not explain how each one is impossible without the other [9].

The main difficulty is the so-called phenomenal consciousness (subjective experience) **(**3.2**)**. The so-called hard problem of consciousness [10]. There is nothing we know more intimately than the conscious experience, but there is nothing more difficult to explain. But this difficult is far from being exclusive to AI. For example, in Neuroscience the farthest one gets are the neural correlates of access consciousness (representations) **(**3.1**)**. In AI the consciousness one gets are also access consciousness, agents have representations of their own internal states, the so-called "self-awareness" **(**3.3**)**. Examples of these agents are Homer – implemented by Vere and Bickmore [11] – and COMA ("Conscious Machine") – project by Schubert, Schaeffer,Hwang, and Haan [12].

Here there is an analogy between the state of art of AI and the state of art of Neuroscience, in both fields of research the consciousness one gets is access consciousness **(**3.1**)**. Architectures such as SOAR http://sitemaker.umich.edu/soar, IDAhttp://csrg.cs.memphis.edu/csrg/html/IDA/ida_home.html and ACT-R http://act-r.psy.cmu.edu are computational models of human cognition (for example real processing time), but researchers working in those areas do not explicitly attempt to build an agent with access consciousness (representations).

Other research projects involve the construction of androids. These have provided an experimental device for various debates. That in which our essay is about, the debate about the relationship between the mind and the body (unifying the psychological and biological), but also the following: the relationship of the social interaction with internal mechanisms (unifying Social Sciences and Cognitive Psychology), reductionism in Neurosciences (the so-called theories of creation of artificial intelligence), connectionism versus modularity in Cognitive Science (the architectures which produce responses similar to human ones), nature versus creation (the relative importance of innateness and learning in social interaction). The construction of androids could very well provide experimental data so that the theories of subjectivity and the body [4] are no longer missing.

Here we must note the following: missing a theory of subjectivity is not missing information ("information" in the sense of the Information Science), this information may be available and still lacking a theory of subjectivity. For example, consider what happens when you look at a Necker cube: suddenly it flips, and although the retinal image and visible 2-D structure are unchanged, the 3-D interpretation is different. Lines, or rather cube-edges, that once sloped down away from the viewer now slope up away from the viewer, and the vertical square cube face that was previously further away is now nearer. The Necker flip in what it's like to see the pattern of lines as a cube is likely to occur in visually sophisticated robots, under appropriate conditions. There can be no reason for which that variation in what it is like to see these lines as a cube could not take place in robots visually sophisticated (in appropriate conditions), but be or not be well informed about this ***x*** it is a epistemological problem, not a ontological problem (and in this sense, of information not being a ontological problem, Aaron can say that there is here – subjectivity - no philosophical problem [13]). We have information (Information Science) but not a theory of subjectivity because here we are confusing two things: epistemology and ontology. One thing is how we know; another what things are. Information Science contributes to the study of human cognition and also to the study of subjectivity: contributes not exhausted the study of subjectivity.

The so-called Turing Test assumed, in its evaluation of intelligence, that the mental does not have to be embodied [14]. But Turing was wrong regarding the nature of the mental. The so-called Total Turing Test (TTT) preserves the idea that the mental has to be embodied [15]. The candidate to the TTT has to be capable of doing, in the world of objects and persons, all they can do and do them in a way that is indistinguishable (to people) from their workings. The environment and the set design [4]. We have good reasons to build androids. By implementing neuron-cognitive mechanisms in androids, evaluating their interactions with human beings, researchers can hope to build a bridge, for example, between (Cognitive) Neuroscience and the Behavioural Sciences: with androids we have an experimental apparatus for tests of subjectivity, our subjectivity being the same as theirs (even if they have not subjectivity, we put our subjectivity in them experimentally).

As Neuroscience has progressed in the study of emotion and cognition, the dualist philosophical viewpoint that the mind is clearly separated from the body becomes less and less credible. The functionalist philosophical point of view that the mind is purely computational and independent of its physical implementation is also increasingly less plausible. The human mind depends a lot more directly on the human brain and body than dualists and functionalists would like. Cognitive Science provides some support for a version of the so-called mind-brain materialism (for a theory free of mental and physical concepts with which we currently operate). In contrast to dualism, Cognitive Science supports the idea that the mind is not a different substance than the material that makes up the body. But it still needs to deal with the problem of consciousness. In Neuroscience neurophysiologic correlates of the consciousness **(**3.1**)** are mentioned but this is still not the so-called phenomenal consciousness **(**3.2**)** of what is it like to be in a mental state (for example, pain) – in contrast to having access to (representations of) that state [1]. But there is a functionalist reply to this objection. Look up.

Functionalism is the viewpoint in which mental states are made up of causal relationships among themselves, with sensory inputs (for example, perceptions) and with behavioural outputs (for example, actions) [16]-[17]. What makes a mental state be the type of mental state that it is (for example, pain, or the belief that lions are dangerous), is the causal relationship with perceptual stimuli, behavioural responses and other mental states. Take note that in accordance with this definition the particular aspect of the mental state, when realized in the human body, remains open.

The reply then is the following. Our brain is embodied in an environment. We have the mind we have because of the body we have (plus the environment). The functionalist can then relativise mental states to species, the so-called type-type

identity theory relativised to species. The idea is that, for example in humans, there would be a real identity of a type of mental state and a certain type of neurophysiologic state, but this does not preclude that in other species, the same kind of mental state (a mental state whose place in the causal network would be the same or approximately the same as that occupied by its counterpart in humans), could be physically realized by another type of physical state (another kind of neurophysiologic state, a electronic state, etc.). Moreover, AI it is a proper part of Cognitive Science. And we are going to conclude.

5 Conclusions

We need both theories, a theory of subjectivity and a theory of the body. Human beings have the mind that has because they have the body that have, there are no disembodied mind, outside the environment (as instantiated by humans). The mind, even for a dualist, gets most of its stimulation from the body. Furthermore, the mind acts through the body. Given that so much of mental activity arises from bodily stimulation and so much of it is designed to contribute to bodily movement, the human mind is radically unlike, say, the mind of a pure intellect as God (if exist). Taking this seriously, it seems that the human mind could not exist without a body, even if the mind is a distinct substance.

The consciousness of human beings is both of access and phenomenal.

Our problem is that there is no place for a necessary connection with physiology in the space of possible development defined by the concept of the mind. Although such conceptual expansion does not imply a contradiction with the essential nature of the subjective experience, nothing precludes a expanded concept of mind from preserving the features of the former concept and allowing the discovery of this connection [9]-[18].

Homer and COMA (only to return to the examples given above) have access consciousness (representations of) but not phenomenal consciousness (subjectivity).

Access consciousness of agents as Homer and COMA cannot be separated from a body (Total Turing Test), but this body cannot be any aggregate of matter, a body must be indistinguishable from humans to humans: human beings looking at these bodies and confuse us, to process them as to other humans. The phenomenological properties of the bodies of these agents, the way they appear to us, being indistinguishable from the phenomenological properties of human bodies. Our brain processes androids (note that the sophisticated robots Aaron talks about have a body very different from ours [13]) as human for two seconds (there are studies showing that in 70% of participants this is the case [19]). It is for this that we need a theory of subjectivity and a theory of the body. By analogy with the concept of sound (for example), we need a conceptual expansion of the concept of the body: a expansion that does not involve conflict with the essential nature of subjective experience of the body.

6 References

[1] Ned Block, *"Some Concepts of Consciousness"*. In Chalmers, David J. (ed.), *Philosophy of Mind: Classical and Contemporary Readings*, pp. 206-218. Oxford University Press, 2002.

[2] Jason Holt, *Blindsight and the Nature of Consciousness*, Broadview Press, Ontario, 2003.

[3] John R.Searle, *"Minds, Brains, and Programs"*,*Behavioral and Brain Sciences*, 3, 1980, pp.417-424.

[4] Helder Coelho, *Teoria da Agência, Arquitectura e Cenografia* [Theory of Agency, Architecture and Set Design], 2008.

[5] Thomas Nagel, *"What is it like to be a bat?"*, *Philosophical Review*, Vol. 83, No. 4, October 1974, pp. 435-450. Reprint Ned Block (ed.), *Readings in Philosophy of Psychology*, Vol. 1, Cambridge, Massachusetts, Harvard University Press, 1980, pp. 159- 168.

[6] Gerald M. Edelman, *Second Nature - Brain Science and Human Knowledge*, Yale University Press, 2006.

[7] John R. Searle, *The Rediscovery of the Mind*, Massachusetts Institute of Technology, 1992.

[8] John R. Searle, *Mind - A Brief Introduction*, Oxford, Oxford University Press, 2004.

[9] Thomas Nagel, *"Conceiving The Impossible and The Mind-Body Problem"*, *Philosophy*, Vol.73, No. 285, July 1998, pp. 337-352.

[10] David J. Chalmers, *"Facing up to the problem of consciousness"*. *Journal of Consciousness Studies*, 2(3):200-219, 1995.

[11] Steven Vere and Timothy Bickmore, *"A basic agent"*, *Comput. Intell.*, 6 (1):41-60, 1990.

[12] Lenhart K. Schubert, Stephanie Schaeffer, Chung Hee Hwang and Johannes de Haan, *EPILOG: The Computational System for Episodic Logic. USER GUIDE*, 1993.

[13] Aaron Sloman, *"What is like to be a rock?"*, 1996. http://www.cs.bham.ac.uk/research/projects/cogaff/misc/like_to_be_a_rock/rock.html

[14] A. M. Turing, *"Computing machinery and intelligence"*, *Mind*, 59, 1950, pp. 433–460.

[15] S. Harnad, *"Other bodies, other minds: A machine incarnation of an old philosophical problem"*,*Minds and Machines*, 1, 1991, pp. 43–54.

[16] Hilary Putnam, *"Psychological Predicates*" (1967), W. H. Capitain and D. Merrill (eds.), *Art, Minds, and Religion*, Pittsburgh, PA, University of Pittsburgh Press, 1967, pp. 37-48. Reprint under the title *"The Nature of Mental States"*, Ned Block (ed.), *Readings in Philosophy of Psychology*, Vol. 1, Cambridge, Massachusetts, Harvard University Press, 1980, pp. 223- 231.

[17] Sydney Shoemaker, "*Some Varieties of Functionalism"*, *Philosophical Topics*, No. 12, 1981, pp. 83-118. Reprint SHOEMAKER, Sydney., *Identity, Cause and Mind*, Cambridge, Cambridge University Press, 1984, pp. 261-285.

[18] Thomas Nagel, *"The Psychophysical Nexus"*, *Concealment and Exposure and Other Essays*, New York, Oxford University Press, 2002.

[19] H. Ishiguro,"*Android science: Toward a new cross-disciplinary framework*". *Cogsci - 2005 workshop: Toward social mechanisms of android science*, pp. 1–6, Stresa, Italy, 2005.

SESSION

ARTIFICIAL NEURAL NETWORKS AND APPLICATIONS

Chair(s)

TBA

Multi-Objective Optimisation in Time Series: Time Delay Agreement

Juan Carlos Cuevas-Tello[1] **and Hector G. Perez-Gonzalez**[1]
[1]Autonomous University of San Luis Potosi, Engineering Faculty, San Luis Potosi, SLP, Mexico
{cuevas,hectorgerardo}@uaslp.mx

Abstract— *Several time delay estimates have been reported for the quasar Q0957+561. They come from distinct data sets and published separately. This paper presents a methodology to estimate a single time delay given several data sets by using multi-objective optimisation. We use General Regression Neural Networks (GRNN) to estimate the time delay, which is one of the most accurate time delay estimators – and faster. For the time delay agreement, we use hill-climbing search. We found that the best agreement for the time delay on Q0957+561 is* $\Delta = 420$ *days.*

Keywords: Neural networks and applications, Heuristic searching methods, Applications: time series in astronomy

1. Introduction

Since it was predicted that the Hubble parameter can be estimated through time delays on gravitational lenses [1], many observation campaigns have been lunch since then [2], [3], [4], [5], [6], and new projects for ambitious surveys like Large Synoptic Survey Telescope (LSST) and the SuperNova Acceleration Probe (SNAP) devoted to study dark matter are in development. Moreover, current surveys like The Sloan Digital Sky Survey (SDSS) and Sloan Lens ACS (SLACS) are generating a tremendous amount of large monitoring data sets. The above surveys are not useful only to estimate the Hubble's parameter, because they are also important to study lensed supernovae (SNe) [7]. Therefore, time delay estimations become a big issue to study dark matter and microlensing.

So far methods to estimate time delays have been used along a monitoring campaign devoted to a single quasar [2], [3]. However, it is important to have a methodology that allows to estimate a time delay if one has several data sets coming from different surveys. In particular, it is well know that given a data set you can estimate a time delay which may be different from another data set, even if they come from the same source. It is the case for the most studied quasar Q0957+561 [2], [3], where a controversy started in the 90's and apparently stopped in 1997 with the work of Kundic et al. [2]. However, the definite time delay is based on a single data set (g-band). After that, more time delay estimates have been published for this quasar, e.g. see [4], [5], [8], [9].

Here we present a methodology to estimate time delay using several data sets from the same source. In particular we study data sets from the quasar Q0957+561. As we said above, each data set may give you a distinct time delay estimation, so we do multi-objective optimisation with hill climbing search. This allows us to have a time delay agreement given several data sets. The idea of multi-objective optimisation is that there is not only a single solution, then there is a set of solutions [10]. We found that the best time delay is $\Delta = 420$ days for the Q0957+561.

According to the best of our knowledge, this is the first approach to deal with this problem in time series via multi-objective optimisation. In practice, astronomers estimate the time delay separately for each pair of time series and by hand they study the time delay agreement for the same quasar.

The reminder of the paper is orginised as follows: the next section describes the data sets used in this research. In §3, we describe the proposed methodology. It follows the experiments and results section, and finally it comes the conclusions and future work.

2. Data sets

We do study six different data sets from the same quasar Q0957+561. The details are in Table 1 and the plots in Fig. 1. For DS1, The whole series was provided by R. Schild [11], private communication. The column labelled n corresponds to the amount of observations per data set. Through the third column, we specify the kind of data, optical o radio, and the Type means the filter and the frequency used to obtain such data sets. The Q0957+561 is a two images quasar, so there is either an offset or a ratio between the two components: image-A and image-B, which correspond to optical and radio data respectively.

3. Methodology

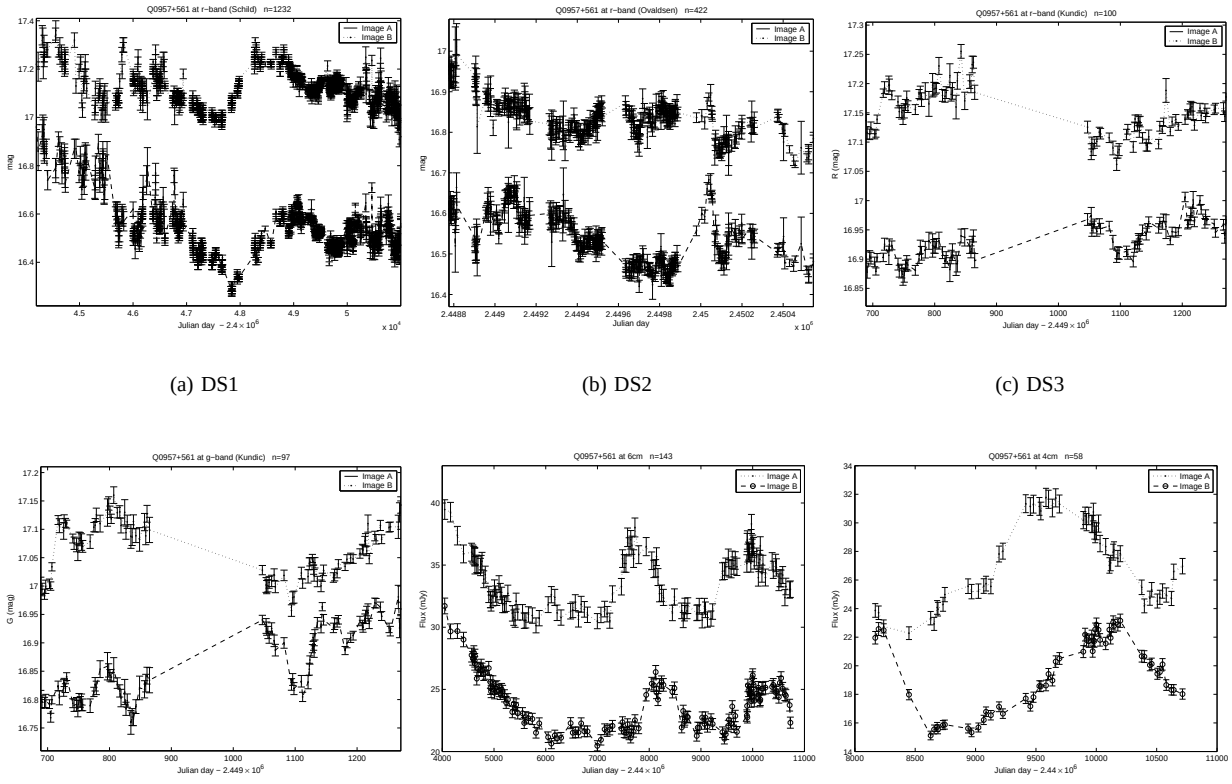
Assume that you have several data sets for the same quasar, which are obtained through a monitoring campaign manually or automatically (e.g. SNAP, LSST, SDSS), we denote them as:

$$d_i \quad i = 1, 2, ..., N \tag{1}$$

where N is the number of data sets, and D contains all data sets d_i. For each data set d_i, we have a time delay estimation

Table 1: Data sets: Q0957+561

Id	n	Data	Type	Ratio/Offset	Monitoring Range	Ref
DS1	1232	optical	r-band	0.05	16/11/1979 – 4/7/1998	[11]
DS2	422	optical	r-band	0.076	2/6/1992 – 8/4/1997	[5]
DS3	100	optical	r-band	0.21	3/12/1994 – 6/7/1996	[2]
DS4	97	optical	g-band	0.117	3/12/1994 – 6/7/1996	[2]
DS5	143	radio	6cm	1/1.43	23/6/1979 – 6-Oct-1997	[3]
DS6	58	radio	4cm	1/1.44	4/10/1990 – 22/9/1997	[3]



(a) DS1 (b) DS2 (c) DS3

(d) DS4 (e) DS5 (f) DS6

Fig. 1: Data sets: Q0957+561. Image-A on DS1 is shifted up 0.6mag for clarity; image-A of DS2 was shifted up 0.25mag; for DS4, the shift up is 0.05mag. For more details on these data sets see Table 1.

Δ_i and also a set of parameters P_i. The size of the set of parameters P_i depend on the method used to estimate the time delay. The best fitting error associated to each Δ_i is denoted by e_i.

The first step is the initialisation of all time delays Δ_i^0 and parameters involved P_i^0, see Algorithm 1.

The next step is obtaining the set of solutions, see Algorithm 2.

What really the Algorithm 2 does, it is to find a set of solutions rather than a single solution. In other words, we are performing multi-objective optimisation.

On one hand, the time delay estimation methods are looking for the best fitting given a time delay Δ and two time series (image-A and image-B for Q0957+561). This works for a single data set, and this is one objective In Algorithm

Algorithm 1: Initialization (D)

for each $d_i \in D$
{
Read file for d_i ;
Estimate Δ_i^0, e_i^0 and P_i^0 ;
}

Algorithm 2: Find set of solutions (D, Δ_i^0, e_i^0, P_i^0)

```
/* MAXG is the # of iterations to optimise E[Δ̄ − Δ_i]                    */
Δ̄ ← E[Δ_i^0] ;
ē ← E[e_i^0] ;
Δ_i ← Δ_i^0 ;
P_i ← P_i^0 ;
g ← 1 ;
plot Δ̄, ē ;
while E[Δ̄ − Δ_i] > 0 and ē > 0 and g <MAXG
{
        for each d_i ∈ D
        {
                Perturb P_i to obtain P̃_i ;
                Get Δ̃_i with P̃_i ;
                /* Hill Climbing search, optimise E[Δ̄ − Δ_i]             */
                if E[Δ̄ − Δ_i] > E[Δ̄ − Δ̃_i]
                {
                        P_i ← P̃_i ;
                        Δ_i ← Δ̃_i ;
                        Δ̄ ← E[Δ_i] ;
                }
        }
        plot E[Δ̄ − Δ_i], ē ;
        g ← g + 1 ;
}
```

2, we refer to the best fitting as e_i, where $\overline{e} = E[e_i] = 1/N \sum_i^N e_i$. So we assume that e_i is calculated when the best time delay Δ_i is estimated. Most of the time delay methods start with a set of suggested time delays, then the best time delay is found when the fitting error e_i is minimal.

On the other hand, here we want that all time delays, for each data set d_i, converge to a single time delay. Remember that the theory predicts that the time delay of the same gravitational lens must be the same, regardless the observation methodology (e.g. optical or radio telescopes). Then, the other objective is to get the minimal $E[\overline{\Delta} - \Delta_i]$, where $E[\overline{\Delta} - \Delta_i] = 1/N \sum_i^N (\overline{\Delta} - \Delta_i)$ and $\overline{\Delta} = E[\Delta_i] = 1/N \sum_i^N \Delta_i$.

Finally, the Algorithm 2 ends when either $\overline{e}$ is zero or $E[\overline{\Delta} - \Delta_i]$ is zero. Otherwise, Algorithm 2 stops when the maximum number of iterations (MAXG) has been reach.

4. Experiments and Results

We have two accurate methods to estimate time delays: i) EA-M-CV method is an evolutionary algorithm with mixed representation (integer and real numbers), and a objective function based on kernel formulation and cross-validation [12]. This method models a single underlying function that generates the two images plus the delay Δ between them. ii) GRNN method, which is based on radial basis functions (RBF) from neural networks theory [13]. As many methods in the literature, GRNN method mixes the two images into a single one given a trial time delay Δ, and when the best fitting e is reached the best time delay comes up (see [14]). The only parameter to estimate with GRNN method is the spread, which is the width of the basis functions ω, Gaussian functions. Therefore, $P_i = \omega_i$.

In practice, EA-M-CV with a population of 300 individuals and 150 generations was used to estimate a time delay on DS4 (see §2. Since EA-M-CV is stochastic, it is necessary to run several realisations, and 10 realisations takes about 3 hours (Matlab program). GRNN takes only 20 minutes (Matlab program), and GRNN with a C program takes only one second. All these experiments with the same data set (DS4) and running on the same machine (MacBook Pro, 2.4Ghz, Intel Core 2 Duo, 4Gb RAM, MacOS-X ver. 10.5.6).

Consequently, all experiments to test the methodology in §3 are performed with GRNN (C program). The results after performing the Algorithm 1 on the data sets in Table 1 are shown in Table 2. We test ω in the range of 5 to 8 with increments of 0.1 units, and trial time delays in the range of 400 to 449 with unitary increments. For perturbing P_i to obtain $\widetilde{P}_i$, in Algorithm 2, we generate random numbers with a Gaussian distribution, zero mean and standard deviation set to 5.

The results after 100 iterations (MAXG=100) are in Table 3. The total elapsed time was 4421 seconds (1 hour, 13

Table 2: Time Delays after initialisation

DS	Δ^0	e	$P_i = \omega_i$
1	428	8.510292e-04	7.3
2	427	2.491830e-04	7.45
3	426	2.112661e-04	7.7
4	420	3.081128e-04	7.1
5	449	9.356649e-02	6.5
6	402	8.736157e-02	8.0
	$\overline{\Delta} = 425.33$	$\overline{e} = 0.03$	$E[\overline{\Delta} - \Delta_i] = 9.0$

Table 3: Final Time Delays after 100 iterations

DS	Δ^0	e	$P_i = \omega_i$
1	422	2.666853e-04	0.84
2	424	1.514531e-04	1.91
3	420	1.052118e-04	2.80
4	420	3.081128e-04	7.10
5	419	4.397565e-03	0.25
6	416	1.792195e-03	0.66
	$\overline{\Delta} = 420.17$	$\overline{e} = 1.17E-3$	$E[\overline{\Delta} - \Delta_i] = 1.5$

minutes). The best time delay by using this methodology is $\Delta = 420$ days.

The set of solutions are depicted in Fig. 2. The first solution, during initialisation stage, is when $E[\overline{\Delta} - \Delta_i] = 9.0$, on the right. The best solution is when $E[\overline{\Delta} - \Delta_i] = 1.5$, the best time delay agreement. Note that in Fig. 2, following the initialisation stage, at the right, the following solutions are getting closer to the best solution in terms of $E[\overline{\Delta} - \Delta_i]$, but the fitting error is increasing. This is the tradeoff in multi-objective optimisation.

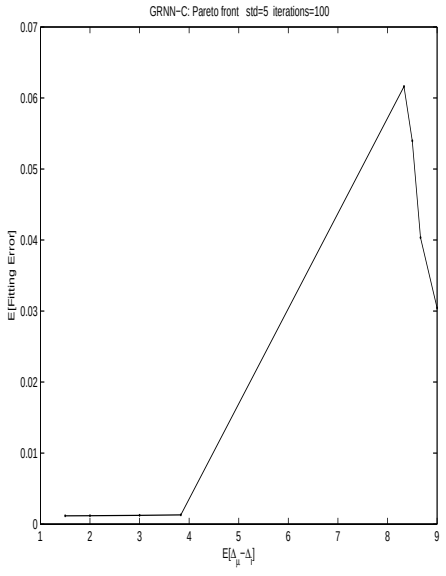


Fig. 2: Set of solutions, starting from the right-hand side to the left-hand side.

5. Conclusions and Future Work

The GRNN method is an accurate and fast method for time delay estimation. The EA-M-CV is robust time delay estimation but it cannot manage to analyse DS1 because the data set is large (more than one thousand observations).

Most of the data sets converge or are close to the time delay of 420 days. The only data set that underestimate this time delay is DS6. In Fig. 3, we can see that in the curves showing where is the best time delay. In Fig 3a, when ω is low, several time delay are feasible: 400, 416 and 420 days because the curve is sharp. However, at the initialisation stage, Δ is 402 with $\omega = 8$ (see Fig. 3c). Now some value in between, that is, when $\omega = 6.5$ we can see that there is some feature at $\Delta = 420$, that is, the agreed time delay for all other data sets (see Fig. 3b).

The set of solutions in Fig. 2 may change since the perturbing method used in the experiments is stochastic (see Algorithm 2). We did several simulations and most of them converge to the solution shown in Fig. 2.

As part of the future work, it is desirable to compare the performance of this methodology with a evolutionary-based methodology, that is, evolutionary computation (EC). In theory, EC is a global optimisation algorithm that may find a better set of solutions.

Another idea is to fix the initial Δ_i^0 into a single time delay Δ^0 (say the mean) and then plug it back into the variable representing the time delay estimation into the model, regardless the time delay estimation method. Then, for each data set obtain the log likelihood. The best time delay it may come with the sum of all likelihoods.

6. Acknowledgements

This paper was possible thanks to Peter Tino and Somak Raychaudhury our collaborators in this topic at University of Birmingham, UK. This work has been partially supported by grants C09-FAI-03-35.35, PROMEP/103.5/08/1696 and PROMEP/103.5/09/4280.

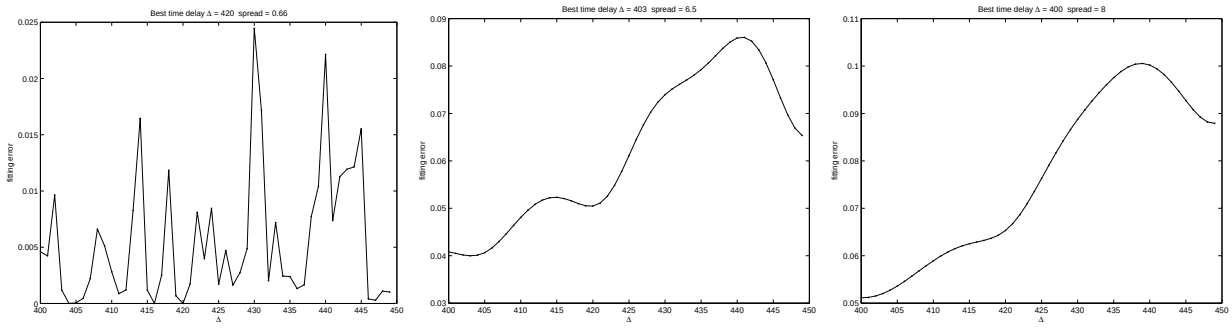
References

[1] S. Refsdal, "On the possibility of determining Hubble's parameter and the masses of galaxies from the gravitational lens effect," *Monthly Notices of The Royal Astronomical Society*, vol. 128, pp. 307–+, 1964.

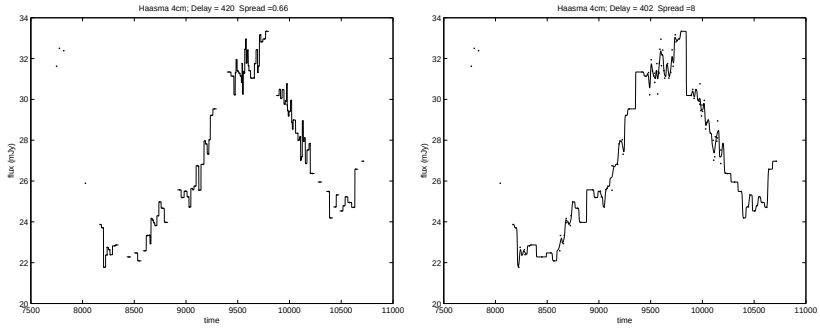
[2] T. Kundic, E. Turner, W. Colley, J. Gott-III, J. Rhoads, Y. Wang, L. bergeron, K. Gloria, D. Long, S. Malhorta, and J. Wambsganss, "A robust determination of the time delay in 0957+561A,B and a measurement of the global value of Hubble's constant," *Astrophysical Journal*, vol. 482, no. 1, pp. 75–82, 1997.

[3] D. Haarsma, J. Hewitt, J. Lehar, and B. Burke, "The radio wavelength time delay of gravitational lens 0957+561," *Astrophysical Journal*, vol. 510, no. 1, pp. 64–70, 1999.

[4] W. Colley, R. Schild, C. Abajas, D. Alcalde, Z. Aslan, I. Bikmaev, V. Chavushyan, L. Chinarro, J. Cournoyer, R. Crowe, V. Dudinov, A. Evans, Y. Jeon, L. Goicoechea, O. Golbasi, I. Khamitov, K. Kjernsmo, H. Lee, J. Lee, K. Lee, M. Lee, O. Lopez-Cruz, E. Mediavilla, A. Moffat, R.Mujica, A. Ullan, J. Munoz, A. Oscoz, M. Park, N. Purves, O. Saanum, N. Sakhibullin, M. Serra-Ricart, I. Sinelnikov, R. Stabell, A. Stockton, J. Teuber, R. Thompson, H. Woo, and A. Zheleznyak, "Around-the-clock observations of the



(a) DS6, $\omega = 0.66$ (b) DS6, $\omega = 6.5$ (c) DS6, $\omega = 8.0$



(d) DS6, $\Delta = 420$, $\omega = 0.66$ (e) DS6, $\Delta = 402$, $\omega = 8$

Fig. 3: DS6: Q0957+561. a)-c) show the curves showing the best time delay. d)-e) depicts the reconstructions of the combined curve given a time delay Δ and $P_i = \omega$.

Q0957+561 A,B gravitationally lensed quasar. II. results for the second observing season," *Astronomy and Astrophysics*, vol. 587, no. 1, pp. 71–79, 2003.

[5] J. Ovaldsen, J. Teuber, R. Schild, and R. Stabell, "New aperture photometry of QSO 0957+561; application to time delay and microlensing," *Astronomy and Astrophysics*, vol. 402, no. 3, pp. 891–904, 2003.

[6] A. Eigenbrod, F. Courbin, C. Vuissoz, G. Meylan, P. Saha, and S. Dye, "COSMOGRAIL: The COSmological MOnitoring of GRAvItational Lenses. I. How to sample the light curves of gravitationally lensed quasars to measure accurate time delays," *Astronomy and Astrophysics*, vol. 436, pp. 25–35, June 2005.

[7] G. Dobler and C. R. Keeton, "Microlensing of Lensed Supernovae," *Astrophysical Journal*, vol. 653, pp. 1391–1399, Dec. 2006.

[8] J. Cuevas-Tello, P. Tiňo, and S. Raychaudhury, "How accurate are the time delay estimates in gravitational lensing?" *Astronomy and Astrophysics*, vol. 454, pp. 695–706, August 2006.

[9] ——, "A kernel-based approach to estimating phase shifts between irregularly sampled time series: an application to gravitational lenses," in *Machine Learning: ECML 2006*, ser. Lecture Notes in Artificial Intelligence (LNAI 4212). Springer-Verlag, Sept. 2006, pp. 614–621.

[10] *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd ed. Springer, 2007.

[11] R. Schild and D. Thomson, "The Q0957+561 time delay from optical data," *Astronomical Journal*, vol. 113, no. 1, pp. 130–135, 1997.

[12] J. Cuevas-Tello, "Estimating time delays between irregularly sampled time series," Ph.D. dissertation, School of Computer Science, University of Birmingham, 2007, http://etheses.bham.ac.uk/88/.

[13] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[14] R. Gonzalez-Grimaldo and J. Cuevas-Tello, "Analysis of time series with neural networks," in *Proceedings of 7th Mexican International Conference on Artificial Intelligence (MICAI)*. IEEE Computer Society, Nov. 2008, pp. 131–137.

FACIAL ACCESS CONTROL BASED ON VG-RAM WEIGHTLESS NEURAL NETWORKS

Jairo Lucas de Moraes, Alberto F. De Souza, Claudine Badue

Departamento de Informática, Universidade Federal do Espírito Santo
Av. Fernando Ferrari, 514 – 29060-970 – Vitória-ES – Brazil

Abstract

We present and evaluate a system for access control that uses only facial biometrics as access key. Our system first detects a face in an image, then tries to recognize the face, and finally decides on grating or not access based on a belief computed during the face recognition process. We use the Viola-Jones approach for face detection, Virtual Generalizing Random Access Memory Weightless Neural Networks (VG-RAM WNN) for face recognition, and Bayesian inference for access control.

We simulated the access control to a given resource for a universe of 50, 100 and 200 users. For the set of 200 users, the system was able to correctly authenticate 93.0% of the users with a False Acceptance Rate (FAR) of only 0.8%; for the set of 100 users, 90.3% of the users with a FAR of 1.8%; and for the set of 50 users, 93.1% of users with a FAR de 4.8%.

Keywords: *Access control, facial biometrics, VG-RAM weightless neural networks, Bayesian inference*

1. Introduction

The process of electronic recognition of the identity of individuals has become increasingly commonplace. Its use goes from the access of garages of buildings via magnetic cards to the login into banking sites via identification numbers or electronic passwords. Today, at various levels of transparency, individuals coexist with automatic recognition processes in their day-to-day activities.

According to Hong and Jain [1], traditional approaches for personal recognition are based on "something that you know", such as a personal identification number, or "something that you have", such as an identification card. Unfortunately, for many applications, these methods may not be secure enough to ensure proper personal recognition, because they lack the capability to differentiate between a genuine individual and an impostor who fraudulently acquires the access privilege.

Biometric systems—which include devices to capture biometric information, such as face images, iris images, fingerprints, etc., and databases and software for storage and management of this information [2]—perform the automatic recognition of individuals based on their physiological and/or behavioral characteristics, that is, the individuals themselves become the "identification key", which makes the process more transparent and less prone to fraud. Due to their wide application in various areas, such as public safety, continuous authentication on computer networks, access control, etc., biometric systems are gaining increasing attention from researchers in academia and industry [3, 4, 5].

Among the various alternative biometric information forms that can now be caught by input devices of personal recognition systems, face images are one of the most convenient. Video capture devices are non-invasive, inexpensive, and easy to use. Moreover, the continuous increase of processor performance over the several last decades allowed the use of more sophisticated, robust, reliable and fast algorithms for face detection and recognition.

In this paper, we evaluate the feasibility of a system for access control using only facial biometrics as access key. In this case, access control would no longer be based on "something that you know" or "something that you have", but on the individual itself. To evaluate the feasibility of an access control system based only on facial biometrics, we developed a prototype of this system that operates fully automatically, being able to detect a face in an image and then perform the recognition of that face, with no human intervention. We use a well known approach proposed by Viola and Jones [6] for face detection and Virtual Generalizing Random Access Memory Weightless Neural Networks (VG-RAM WNN) for face recognition [7, 8]. Lastly, we employed Bayesian inference for the access control decision process.

Many techniques for face recognition have been proposed in the literature [4, 5]. However, instead of the access control problem, most of them have been employed to address: the *face identification problem*, where the system always returns a face that has the most similar features to the input face, even though the input face is not in the knowledge base; or the *face verification problem*, where the system receives an identification number along with facial biometric data and reports whether or not it belongs to the claimed identification number. Different

from the approaches presented in [4, 5], we employ a face recognition technique to address the *access control problem* using only facial biometrics as access key; in this case, the system receives facial biometric data only and reports whether or not it belongs to a user that has access to a particular resource or environment.

Although there are currently face based access control systems commercially available, as far as we could examine in the literature, the combination of techniques we have employed to tackle the problem is unique, and the results we have obtained are promising. Using our prototype, the access control to a given resource was simulated for 50, 100 and 200 users. For the set of 200 users, the system was able to correctly authenticate 93.0% of the users with a False Acceptance Rate (FAR) of only 0.8%; for the set of 100 users, the system was able to correctly authenticate 90.3% of the users with a FAR of 1.8%; and for the set of 50 users, the system correctly authenticated 93.1% of users with a FAR of 4.8%.

This paper is organized as follows. After this introduction, in Section 2 we present our prototype of an access control system based only on facial biometrics. In Section 3, we describe our experimental methodology, in Section 4, we analyze our experimental results and, in Section 5, we discuss them. Our conclusions follow in Section 6.

2. Access Control Based on Facial Biometrics

We developed a prototype of an access control system based only on facial biometrics. Our system operates fully automatically in three steps: (i) detection of a face in an image; (ii) recognition of the detected face; and (iii) Bayesian inference for determining if the access should be granted. In the first step, given an arbitrary image, the system determines whether or not there are faces in the image and, if so, it returns the image location and extent of each face. In the second step, given a detected face, the system returns the most similar face, among those enrolled in the knowledge base, along with a matching score, that quantifies the similarity between the detected face and the most similar face in the knowledge base. In the third step, given the matching score and using the Bayes' rule, the system computes a probability measure that indicates the degree of belief of the system in that the detected face belongs to an individual with granted access.

The system final decision is regulated by a threshold: if the degree of belief of the system in that the detected face belongs to an individual with granted access is less than the threshold, than he/she is rejected as an impostor; otherwise, he/she is accepted as an individual with granted access (or genuine individual for short).

In the following, we describe each of these three steps.

2.1. Face Detection

We use the well known object detection technique proposed by Viola and Jones [6] for the task of face detection. This technique uses integral images for fast feature extraction, AdaBoost [9] for classification, and a method for combining the classifiers in a cascade, which allows background regions of the images to be quickly discarded while spending more computation on promising face-like regions.

We have used the Viola-Jones approach to detect faces and also the eyes within the faces. The knowledge of the eyes' position is important for proper face recognition, since it allows a more precise reference for the face recognition system to operate. We found the correct detection of the eyes hard to obtain in some cases. Because of that, our face detection sub-system tries and recognizes the cases were it was not possible to correctly detect the eyes and, in such cases, approximates their position as a previously computed average position.

2.2. Face Recognition

We use Virtual Generalizing Random Access Memory Weightless Neural Networks (VG-RAM WNN) [10, 11] for face recognition. VG-RAM WNN is an effective machine learning technique that offers simple implementation and fast training and test [10]. In previous works [7, 8], we evaluated the performance of VG-RAM WNN on face recognition using well known face databases. Our experimental results showed that, even when training with a single face image per individual, VG-RAM WNN are robust to various facial expressions, occlusions and illumination conditions, showing better performance than many well known face recognition techniques. This has motivated us to use VG-RAM WNN for the face recognition step of our access control system.

2.2.1. VG-RAM WNN

RAM-based neural networks, also known as *n*-tuple classifiers or weightless neural networks, do not store knowledge in their connections but in Random Access Memories (RAM) inside the network's nodes, or neurons. These neurons operate with binary input values and use RAM as lookup tables: the synapses of each neuron collect a vector of bits from the network's inputs that is used as the RAM address, and the value stored at this address is the neuron's output. Training can be made in one shot and basically consists of storing the desired output in the address associated with the input vector of the neuron [12].

In spite of their remarkable simplicity, RAM-based neural networks are very effective as pattern recognition tools, offering fast training and test, in addition to easy implementation [10]. However, if the network input is too large, the memory size becomes prohibitive, since it must be equal to 2^n, where n is the input size. Virtual Generalizing RAM (VG-RAM) Weightless Neural Networks (WNN) are RAM-based neural networks that

only require memory capacity to store the data related to the training set [11]. In the neurons of these networks, the memory stores the input-output pairs shown during training, instead of only the output. In the test phase, the memory of VG-RAM WNN neurons is searched associatively by comparing the input presented to the network with all inputs in the input-output pairs learned. The output of each VG-RAM WNN neuron is taken from the pair whose input is nearest to the input presented—the distance function employed by VG-RAM WNN neurons is the Hamming distance. If there is more than one pair at the same minimum distance from the input presented, the neuron's output is chosen randomly among these pairs.

Figure 1 shows the lookup table of a VG-RAM WNN neuron with three synapses (X_1, X_2 and X_3). This lookup table contains three entries (input-output pairs), which were stored during the training phase (entry #1, entry #2 and entry #3). During the test phase, when an input vector (input) is presented to the network, the VG-RAM WNN test algorithm calculates the distance between this input vector and each input of the input-output pairs stored in the lookup table. In the example of Figure 1 the Hamming distance from the input to entry #1 is two, because both X_2 and X_3 bits do not match the input vector. The distance to entry #2 is one, because X_1 is the only non-matching bit. The distance to entry #3 is three, as the reader may easily verify. Hence, for this input vector, the algorithm evaluates the neuron's output, *Y*, as class 2, since it is the output value stored in entry #2.

Lookup Table	X_1	X_2	X_3	Y
entry #1	1	1	0	label 1
entry #2	0	0	1	label 2
entry #3	0	1	0	label 3
	↑	↑	↑	↓
input	1	0	1	**label 2**

Figure 1: VG-RAM WNN neuron lookup table

2.2.2. Face Recognition with VG-RAM WNN

Our VG-RAM WNN architecture for face recognition has a single bidimensional array of $m \times n$ neurons, *N*, where each neuron, $n_{i,j}$, has a set of synapses, $W = (w_1,w_2,...w_{|W|})$, which are connected to the network's bidimensional input, Φ, of $u \times v$ inputs (see Figure 2 and Figure 3). The synaptic interconnection pattern of each neuron $n_{i,j}$ follows a bidimensional Normal distribution with variance σ^2 centered at φ_{μ_k,μ_l} where $\mu_k = \frac{i.u}{m}$ and $\mu_i = \frac{j.v}{n}$; i.e., the coordinates *k* and *l* of the elements of Φ to which $n_{i,j}$ connects via *W* follow the probability density functions:

$$\omega_{\mu_k,\sigma^2}(k) = \frac{1}{\sigma\sqrt{2\Pi}} e^{\frac{-(k-\mu_k)^2}{2\sigma^2}}$$

$$\omega_{\mu_l,\sigma^2}(l) = \frac{1}{\sigma\sqrt{2\Pi}} e^{\frac{-(l-\mu_l)^2}{2\sigma^2}}$$

where σ is a parameter of the architecture (Figure 2). This synaptic interconnection pattern mimics that observed in many classes of biological neurons [13], and is created when the network is built and does not change afterwards.

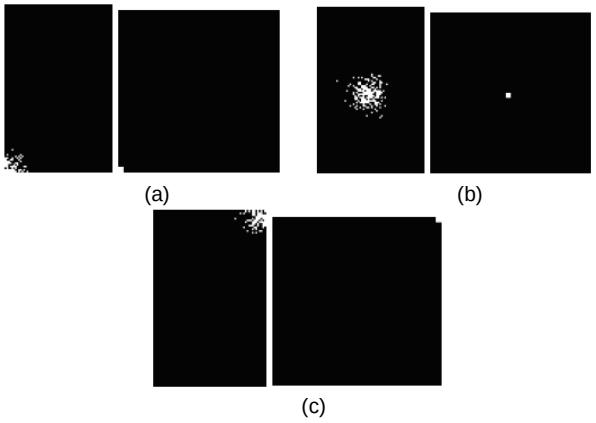


Figure 2: The synaptic interconnection pattern of our VG-RAM WNN architecture for face recognition. (a) Left, input Φ: in white, the elements $\varphi_{k,l}$ of the input Φ that are connected to neuron $n_{1,1}$ of *N* via $\Omega_{1,1,\sigma}(W)$. Right, neuron array *N*: in white, the neuron $n_{1,1}$ of *N*. (b) Left: in white, the elements $\varphi_{k,l}$ of Φ connected to $n_{\frac{m}{2},\frac{n}{2}}$ via $\Omega_{\frac{m}{2},\frac{n}{2},\sigma}(W)$. Right: in white, the neuron $n_{\frac{m}{2},\frac{n}{2}}$ of *N*. (c) Left: in white, the elements of Φ connected to $n_{m,n}$ via $\Omega_{n,n,\sigma}(W)$. Right: in white, the neuron $n_{m,n}$.

VG-RAM WNN synapses can only get a single bit from the input. Thus, in order to allow our VG-RAM WNN to deal with images, in which a pixel may assume a range of different values, we use *minchinton cells* [14]. In the proposed VG-RAM WNN architecture, each neuron's synapse, w_t, forms a minchinton cell with the next, w_{t+1} ($w_{|W|}$ forms a minchinton cell with w_1). The type of the minchinton cell we have used returns 1 if the synapse w_t of the cell is connected to an input element, $\varphi_{k,l}$, whose value is larger than the value of the element $\varphi_{r,s}$ to which the synapse w_{t+1} is connected, i.e., $\varphi_{k,l} > \varphi_{r,s}$; otherwise, it returns zero (see the synapses w_1 and w_2 of the neuron $n_{m,n}$ of Figure 3).

The input face images, *I*, of $\xi \times \eta$ pixels (Figure 3) must be transformed in order to fit into the network's input, Φ. The images are rotated, scaled, and cropped (Figure 4) automatically in three steps: (i) the position of the face in the image is found; (ii) based on the face position, the positions of the eyes are found (Figure 4(b));

and (iii) based on the positions of the face and eyes, the image is rotated, scaled and cropped to fit into Φ. Before being copied to Φ, the transformed image is filtered by a Gaussian filter to smooth out artifacts produced by the transformations (Figure 4(c)).

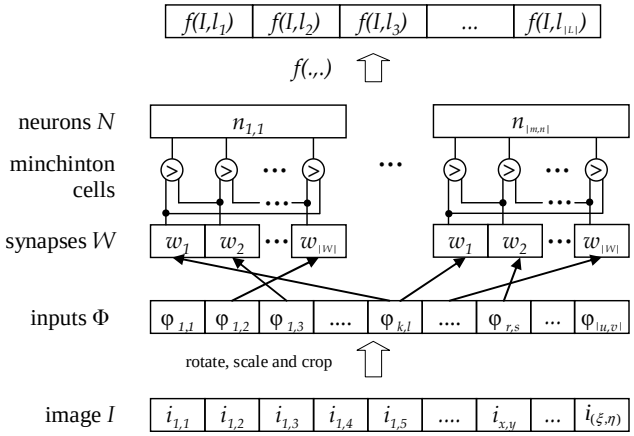


Figure 3: Schematic diagram of our VG-RAM WNN architecture for face recognition

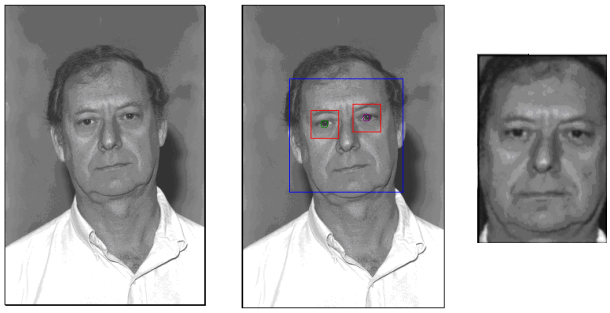


Figure 4: Face image and its preprocessing. (a) Original image; (b) positions of the face and eyes in the image; and (c) rotated, scaled, cropped and filtered image.

During training, the face image I_x of a person p is transformed and filtered, its pixels are copied to the VG-RAM WNN's input Φ, and all $n_{i,j}$ neurons' outputs are set to the value of the label $l_p \in L=\{l_1, \ldots l_{|L|}\}$ associated with the face of the person p ($|L|$ is equal to the number of known persons). All neurons are then trained to output this label with this input image. This procedure is repeated for all images I_x of the person p and, likewise, for all persons in the training dataset. During testing, each face image I_y is also transformed, filtered, and copied to the VG-RAM WNN's input Φ. Then, all neurons' outputs are computed and the number of neurons outputting each label $l_p \in L=\{l_1, \ldots, l_{|L|}\}$ is counted. The network output is given by the label l_p with the largest count along with the percentage of neurons that presented l_p as output for the face image I_y. This percentage is a matching score, $f(I_y, l_p)$, which quantifies the similarity between the face image I_y and the most similar one in the knowledge base that is indexed by the label l_p.

2.3. Access Control

We employed Bayesian inference for addressing the problem of access control. Given a face image I_y, our access control system maps the matching score $f(I_y, l_p)$—that quantifies the similarity between the face image I_y and the most similar image in the knowledge base, indexed by label l_p (Section 2.2.2)—into a probability measure, which indicates the degree of belief of the system in that the face image I_y belongs to a genuine individual. The system final decision is regulated by a threshold for the probability measure: if the probability measure is smaller than the threshold, the user associated with the face image I_y is rejected as an impostor; otherwise, the user is accepted as a genuine individual. The value of the threshold can either be specified by the system operator or automatically tuned using a validation dataset (not a part of the training dataset or the test dataset [15]), by varying the value of the threshold until the performance of the access control system is optimized on the validation dataset. The probability measure is computed using the Bayes' rule as described in the following.

The probability measure of interest, $p(A|B)$, is computed as the probability that a given face image, I_x, belongs to a genuine individual ($p(A)$), given that the neural network returned a matching score, $f(I_x, l_p)$, within an interval $b_i \in B=\{b_1, \ldots, b_{|B|}\}$ ($p(B)$). The random variable A may take two values: 1, if the given face image I_x belongs to a genuine individual; or 0, if the given face image I_x belongs to an impostor. The random variable B may take a continuous value within one of the intervals of $B=\{b_1, \ldots, b_{|B|}\}$.

The probability $p(A|B)$ can be computed using the Bayes' rule, i.e.:

$$p(A \mid B) = \frac{p(B \mid A) \times p(A)}{p(B)}.$$

The probability $p(A)$ can be estimated as the percentage of genuine individuals in the **training and validation datasets**. The probability $p(B)$ can be estimated as the percentage of times the neural network outputs a matching score within each interval of $B=\{b_1, \ldots, b_{|B|}\}$ for images in the **validation dataset**. Finally, the probability $p(B|A)$ can be estimated (using the **validation dataset**) as the percentage of matching scores within each interval $b_i \in B=\{b_1, \ldots, b_{|B|}\}$, given that the network returned genuine individuals.

3. Experimental Methodology

To evaluate our access control system we have used the Color Face Recognition Technology (FERET) database (http://face.nist.gov/colorferet). For that, we divided it into several datasets and used them to train, validate and test the performance of our system according to widely used biometric performance metrics.

3.1. Datasets

The Color FERET database contains a total of 11,338 face images with 512 by 768 pixels. They were collected by photographing 994 individuals at various angles over the course of 15 sessions between the years of 1993 and 1996. Among 13 different poses available in the database (frontal face and head turned from 15 to 75 degrees right and left), we considered 2 frontal face images of 991 individuals: the regular frontal face image, named *fa* in the database, and the alternative frontal face image, *fb*, taken shortly after the corresponding *fa* image. Figure 5 shows the regular frontal face image, *fa*, and the alternative frontal face image, *fb*, of one individual of the Color FERET database.

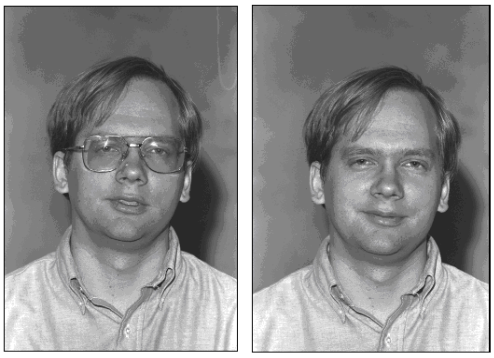


Figure 5: Regular frontal face image, *fa*, and alternative frontal face image, *fb*, of one individual of the Color FERET database

We derived three datasets using the frontal face images *fa* and *fb* of 991 individuals of the FERET database, namely CA1, CA2, e CA3.

3.1.1. CA1 Dataset

To obtain CA1, we partitioned the full dataset (*fa* and *fb* of 991 individuals of the FERET database) into 10 subsets of 100 individuals (the last subset has only 91 individuals). The first of these subsets was further partitioned into training and validation subsets—the training subset comprises the *fa* images of the first 50 individuals, while the validation subset comprises the *fb* images of all 100 individuals. Each of the remaining 9 subsets was partitioned into training and test subsets—the training subset comprises the *fa* images of the first 50 individuals, and the test subset comprises the *fb* images of all 100 individuals of the subset (91 in the last subset).

3.1.2. CA2 Dataset

To obtain CA2, we partitioned the full dataset into 5 subsets of 200 individuals (the last subset has only 191 individuals). The first of these subsets was partitioned into training and validation subsets—the training subset comprises the *fa* images of the first 100 individuals, while the validation subset comprises the *fb* images of all 200 individuals. Each of the remaining 4 subsets was partitioned into training and test subsets—the training subset comprises the *fa* images of the first 100 individuals, and the test subset comprises the *fb* images of all 200 individuals of the subset (191 in the last subset).

3.1.3. CA3 Dataset

To obtain CA3, we partitioned the full dataset into 2 subsets; the first with 400 individuals and the second with 591 individuals. The first of these 2 subsets was partitioned into training and validation subsets—the training subset comprises the *fa* images of the first 200 individuals, while the validation subset comprises the *fb* images of all 400 individuals. The second was partitioned into training and test subsets—the training subset comprises the *fa* images of the first 200 individuals and the test subset comprises the *fb* images of all remaining individuals in the subset.

3.2. Metrics

We evaluated the performance of our access control system according to two standard metrics for biometric recognition systems [16]:

- False Acceptance Rate (FAR), which is defined as the probability of an impostor being accepted as a genuine individual. It can be estimated as the ratio between the number of false positives and the total number of negatives (true negatives plus false positives).
- False Reject Rate (FRR), which is defined as the probability of a genuine individual being rejected as an impostor. It can be estimated as the ratio between the number of false negatives and the total number of positives (true positives plus false negatives).

There is a tradeoff between FAR and FRR. A larger FAR leads to a smaller FRR, while a larger FRR leads to a smaller FAR. In fact, both FAR and FRR are functions of the system threshold *t*. On the one hand, if *t* is decreased to make the system more tolerant, then FAR increases (and FRR decreases); on the other hand, if *t* is increased to make the system more secure, then FRR increases (and FAR decreases). The tradeoff between FAR and FRR is usually depicted in a receiver operating characteristic (ROC) curve, which is a plot of FAR against (1 - FRR) for various threshold values.

4. Experimental Results

In this section, we present the experiments employed to evaluate experimentally the performance of our access control system with 50, 100 and 200 users. To run these experiments, we used the Viola-Jones implementation that is part of the OpenCV library (http://sourceforge.net/projects/opencvlibrary/) and publicly available training datasets for face and eyes detection. We have set the parameters of our VG-RAM WNN to the best values obtained in previous works [8].

In order to use our system, its is necessary to estimate the values of the terms of the Bayes' rule ($p(A)$, $p(B)$ and $p(B|A)$), and to select a threshold for $p(A|B)$ (see Section 2.3). To estimate the values of the terms of the Bayes' rule and to select a threshold for $p(A|B)$ for the case of 50 users, we used the training and validation subsets of the first subset of the CA1 dataset.

In the first subset of the CA1 dataset, the number of users (genuine individuals) is equal to the size of the training subset: 50 individuals; while the number of non-users (impostors) is equal to the size of the validation set minus the number of users: 100 - 50 = 50 individuals. Therefore, $p(A)$ (the probability that a given face image, I_x, belongs to a genuine individual) is equal to 0.5 (the number of users divided by the total number of individuals: 50/100).

To obtain an estimate of $p(B)$ with the first subset of CA1, we train the network with its training set, examine the network output with the validation set, and compute the percentage of times the neural network outputs a matching score within each interval of $B=\{b_1, \ldots, b_{|B|}\}$.

To estimate $p(B|A)$, we compute the percentage of matching scores within each interval $b_i \in B=\{b_1, \ldots, b_{|B|}\}$ for which the network returned genuine individuals.

To select a threshold for $p(A|B)$, we varied its value, plotted a ROC curve and chose a threshold that gives acceptable values of FAR and FRR. The graph in Figure 6 shows the ROC curve of the first subset of CA1 for the threshold values shown in Table 1.

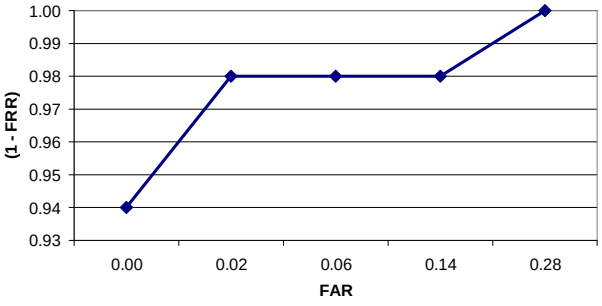


Figure 6: ROC curve of the first subset of CA1

As the graph in Figure 6 shows, our system can achieve a FAR equal to 0% with a FRR of 6% ((1 - FRR) = 0.94) for a threshold of 0.70 (see first data line of Table 1). For a threshold of 0.50, our system can achieve a FAR equal to 2% with a FRR of 2% (a rather small Equal Error Rate - EER).

Table 1: Effect of threshold on FAR and (1 - FRR)

Threshold	FAR	(1 - FRR)
0.70	0.00	0.94
0.50	0.02	0.98
0.35	0.06	0.98
0.25	0.14	0.98
0.09	0.28	1.00

Figure 7 presents the results in Table 1 in graph form (Figure 7(a)), together with equivalent results for CA2 ((Figure 7(b)) and CA3 (Figure 7(c)). As the graphs of Figure 7 show, with 50 users, our system presents an ERR of ~2%, with 100 users an ERR of ~9%, and with 200 users an ERR of ~9% as well.

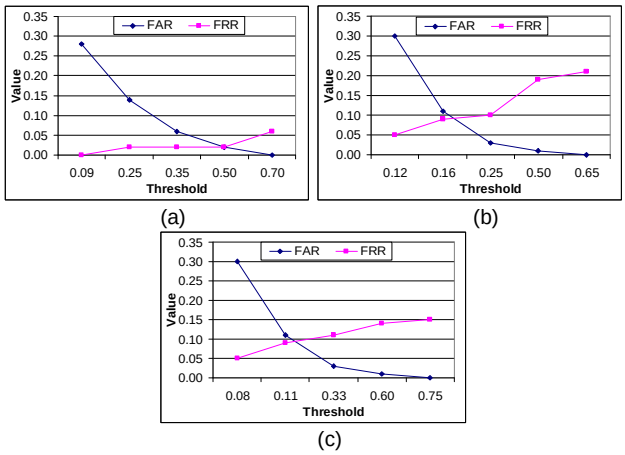


Figure 7: FAR and FRR of the first subset of CA1 (a), CA2 (b) and CA3 (c).

Table 2 presents the performance of our system for 50, 100, and 200 users obtained summarizing the performance derived from the 9 pairs of training and test subsets of CA1, the 4 pairs of training and test subsets of CA2, and the training and test subset of CA3. In order to obtain the results of Table 2, we run experiments with the validation sets of CA1, CA2 and CA3 to try and find the thresholds that favor better (smaller) FAR. In most cases of access control, this is the case, i.e., one prefers a smaller false acceptance rate even if it increases the false rejection rate (or decreases (1 - FRR)).

Table 2: Performance of the access control system for CA1, CA2 and CA3

Data Set	Threshold	FAR	(1 - FRR)
CA1	0.50	0.048	0.931
CA2	0.25	0.018	0.903
CA3	0.33	0.008	0.930

As Table 2 shows, with 50 users, the system correctly authenticated 93.1% of users with a FAR of 4.8%. With 100 users, the system was able to correctly authenticate

90.3% of the users with a FAR of 1.8%. Finally, with 200 users, the system was able to correctly authenticate 93.0% of the users with a FAR of only 0.8%.

5. Discussion

An important aspect of our system is that, in order to select the threshold for $p(A|B)$, it is necessary to use a set of impostors. Actually, the ratio between the size of this set and the size of the set of genuine individuals affects the FAR and FRR of the system. In our experiments we used a ratio equal 1, but this ratio must be estimated for each possible scenario of use of the system in order to properly select a threshold for $p(A|B)$.

Although currently there are many commercially available systems for access control via face recognition, we could only find a single paper about the subject in the literature, [17]. However, the access control system via face recognition proposed by Bryliuk and Starovoitov in [17] employs standard weighted multilayer Perceptron neural networks, instead of weightless neural networks, and train with several image samples per individual. Also, they do not use Bayesian inference to decide about grating the access. Finally, their best FAR and FRR for EER is ~12% with a rather smaller face dataset—40 individuals (http://www.cam-orl.co.uk/facedatabase.html).

6. Conclusions and Future Work

We present a facial access control system based on VG-RAM weightless neural networks (WNN). Our system uses the Viola-Jones approach to detect faces and eyes within these faces in images, forward positions of detected faces and eyes to VG-RAM WNN for recognition of previously trained faces, and employs Bayesian inference for granting or not access to a given resource or environment.

We evaluated our system using the Color FERET database in scenarios that simulate the use of our system with 50, 100 and 200 enrolled users. Our experimental results show that, tuning the system to favor a better (smaller) False Acceptance Rates (FAR), in the case of 50 users, it is able to correctly authenticate 93.1% of users with a FAR of 4.8%. With 100 users it can correctly authenticate 90.3% of the users with a FAR of 1.8%, and with 200 users it can correctly authenticate 93.0% of them with a FAR of only 0.8%.

As future work we plan to deploy our system in a real case scenario and examine its performance using live video.

7. Acknowledgements

We would like to thank Conselho Nacional de Desenvolvimento Científico e Tecnológico-CNPq-Brasil (grants 309831/2007-5, 620185/2008-2, 314485/2009-0) and Fundação de Amparo à Pesquisa do Espírito Santo-FAPES-Brasil (grant 48511579/2009) for their support to this research work.

8. References

[1] L. Hong and A. Jain. Integrating Faces and Fingerprints for Personal Identification. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12:30-36, 1998.

[2] R. Vetter. Authentication By Biometric Verification. Computer, 43(3):28-29, 2010.

[3] M.H. Yang, D.J. Kriegman, and N. Ahuja. Detecting Faces In Images: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(1):34-58, 2002.

[4] W. Zhao, R. Chellappa, J. Phillips, and A. Rosenfeld. Face Recognition: A Literature Survey. ACM Computing Surveys, 35(4):399-458, 2003.

[5] A.S. Tolba, A.H. El-Baz, and A.A. El-Harby. Face Recognition: A Literature Review. International Journal of Signal Processing, 2(2):88-103, 2005.

[6] P. Viola and M. Jones. Robust Real-Time Object Detection. In Proceedings of the 2nd International Workshop on Statistical and Computational Theories of Vision - Modeling, Learning, Computing and Sampling, p. 1-25, Vancouver, Canada, 2001.

[7] A.F. De Souza, C. Badue, F. Pedroni, E. Oliveira, S. S. Dias, H. Oliveira, and S. F. Souza. Face Recognition With VG-RAM Weightless Neural Networks. Lecture Notes in Computer Science, 5163(1):951-960, 2008.

[8] A.F. De Souza, C. Badue, F. Pedroni, S.S. Dias, H. Oliveira, and S. F. de Souza. VG-RAM Weightless Neural Networks for Face Recognition. In: Face Recognition, p. 171-186, InTech, 2010.

[9] Y. Freund and R.E. Schapire. A Decision-Theoretic Generalization of On-line Learning and An Application to Boosting. In Proceedings of the Second European Conference on Computational Learning Theory (EuroCOLT '95), p. 23-27, 1995.

[10] I. Aleksander. From WISARD to MAGNUS: A Family of Weightless Virtual Neural Machines. In: RAM-Based Neural Networks, p. 18-30, World Scientific, 1998.

[11] T.B. Ludermir, A.C.P.L.F. Carvalho, A.P. Braga, and M.D. Souto. Weightless Neural Models: A Review of Current and Past Works, Neural Computing Surveys 2: 41-61, 1999.

[12] I. Aleksander. Self-Adaptive Universal Logic Circuits. IEE Electronic Letters, 2(8): 231-232, 1966.

[13] E.R. Kandel, J.H Schwartz, and T.M. Jessell. Principles of Neural Science, Prentice-Hall International Inc, 2000.

[14] R.J. Mitchell, J.M. Bishop, S.K. Box, and J.F Hawker. RAM - Based Neural Networks, World Scientific, chapter Comparison of Some Methods for Processing Grey Level Data in Weightless Networks, p. 61-70, 1998.

[15] F. Sebastiani. Machine Learning in Automated Text Categorization. ACM Computing Surveys, 34(1):1-47, 2002.

[16] A. Jain, A. Ross, and S. Prabhakar. An Introduction to Biometric Recognition. IEEE Transactions on Circuits and Systems for Video Technology, 14(1):4-20, 2004.

[17] D. Bryliuk and V. Starovoitov. Access Control by Face Recognition Using Neural Networks and Negative Examples. In Proceedings of the 2nd International Conference on Artificial Intelligence (ICAI'2002), p. 428-436, 2002.

Application of Artificial Intelligence in Classification of Maritime Targets

M.R.M. Rizk, H.A. Khater, and M.M. Abdelwahab
Electrical Engineering Department, Alexandria University, Faculty of Engineering , Alexandria, Egypt

Abstract - *Fast classification of a target according to RCS signals is important for many applications. In this paper, we describe the use of the Neural Networks for object classification using collected RCS real data from radar system. Our collected RCS polar plots for 3 ship classes are applied to NNs. This paper proposes three models of three layered feed-forward Neural Network and back-propagation training algorithm. In the first one, we feed 75 inputs to the NN which are frequency, polarization, 72 RCS values and the mean of these values. This method gives 100% overall correct classification. In the second one, we feed four inputs to the NN which are frequency, polarization, aspect angle and its corresponding RCS value. This method gives 76% overall correct classification but it is more efficient in actual scenario because there is no guarantee to view a full revolution of the object. In the third model we introduce six inputs to the NN which are the same inputs of the second model besides adjacent aspect angle and its corresponding RCS value, consequently we obtained 84% overall correct classification. We can use this model when the whole RCS of the object is unavailable.*

Keywords: Radar Cross Section, Neural Networks

1 Introduction

In recent years many methods have been used for target classification, which include image [1-4], sonar signals radiated by ships and underwater targets [5,6] and reflected radar signals [7-13]. Most of them utilized neural networks (NNs) because it proved to be a reliable classification tool especially in the image and signal processing methods. Neural networks are used for different targets classification such as ground targets classification [7]. The target is a ground vehicle in which the features from Forward Scattering Radar signal were extracted manually prior to classification process, using Multi-Layer Perceptron (MLP) back-propagation neural network trained with three back-propagation algorithms. Two types of classifications were analyzed. The first one is to classify the exact type of vehicle, four vehicle types were selected. The second objective was to group vehicles into different categories. The proposed NN architecture was compared to the K-Nearest Neighbor (K-NN) classifier and the performance was evaluated, the results showed that, the proposed NN provides a higher percentage of successful classification than the K-NN classifier. Also naval target classification by R. Soleti [8] uses NN for polarimetric radar target classification. This study showed two different types of feed-forward NNs which have been adopted in order to classify the target echo. The networks used have been tested on two types of simulated targets: A small tonnage ship with a low level of details and medium tonnage ship with higher details. This study found that MLP architecture in general provides better results than Self Organizing Maps (SOM). In [9-11], they demonstrated the applications of the NNs in classifying radar target which used the noisy spectral responses from different types of aircrafts to train the networks. Their performance are compared with the conventional minimum distance classifier for noisy systems and the NN was found to provide better performance in target classification as compared to a conventional scheme. In 2010 a new approach was introduced in [13], which uses support vector machine (SVM) for classification of simple simulated targets (spheres, cylinders, polygons and frusta) according to their synthesized radar cross section (RCS) returns. The results are compared with a Bayesian approach and show that SVM is faster than the Bayesian approach. Also unlike the Bayesian network the SVM can accept very high-dimensional input vectors without preprocessing. In our paper the problem is more realistic, with the new classification algorithm, using collected RCS returns from three real complex targets of ships of different classes (class A, class B and class C). Class B is slightly larger than class A with different details and class C has the largest size with different details. Our collected data are in the frequency range (2-18 GHZ) with two types of polarization: vertical and horizontal. These data are applied to one of the most widely studied and used artificial intelligence classification tool which is feed-forward multilayer Neural Networks [14]. Neural networks could be very fast for RCS pattern recognition because once the network is trained, then the evaluation simply consists of just a few vector-matrix multiplications and some function applications to vectors. Neural networks also have the advantage that significant preprocessing or feature extraction of the RCS signatures may not necessarily be required [13]. In the following sections we will discuss RCS analysis. Then we discuss the proposed neural network structure for RCS signature classification. Then we discuss the results and conclusion. Finally, we propose the future work.

2 Radar Cross Section Analysis

This section deals with RCS parameters and the basic factors related to RCS and its application in maritime targets classification. According to Skolnik [15], RCS is defined as "The radar cross section of a target is the (fictional) area intercepting that amount of power which, when scattered equally in all directions, produces an echo at the radar equal to that from the target.". The term 'area' refers to the unit being m². 'Fictional' means that RCS can actually be much larger than the reflective surface. Also according to [16], it is defined as "Measure of the reflective strength of a target." and it is equal to the power reflected back to the radar divided by power density of the wave striking the target. RCS is denoted by the Greek letter σ and measured in m². RCS is defined as:

$$\sigma = 4\pi \cdot \frac{p_s}{p_i} \qquad (1)$$

Where *Pi*: = power density, or intensity, of a plane wave striking the target, (W/m²).
Ps: = power per unit solid angle reflected by the target, (W).
RCS has a wide spread ranging from 10^{-5} m² for small insects to 10^{6} m² for large ships. Hence, RCS is often stated in the logarithmic decibel scale:

$$\sigma_{dBsm} = 10 \cdot \log\left(\sigma_{sm}\right) \qquad (2)$$

The last equation (2) shows that RCS values (σ_{dBsm}) are expressed in dBsm - decibels relative to a square meter - where dBsm is a direct function of the logarithm to the base ten of the RCS of a target expressed in square meters (σ_{sm}). RCS depends on many factors. These factors are:

1. Position of transmitter/receiver relative to the target, in which the grazing angle (Ø) is the angle measured in the vertical plane between the ray and a reflecting surface as shown in figure 1.
2. Target geometry and material composition.
3. Angular orientation of the target relative to transmitter /receiver, the aspect of a target (α) is its orientation to the axis of the radar beam as shown in figure 1. The nearer the angle between the target axis and the radar beam axis is 90°, the greater is the strength of the echo returned to the antenna.
4. Frequency or wavelength of the radar.
5. Antenna polarization of the radar.

For *simple objects*, such as flat rectangular plates, cylinders and spheres, the RCS can be calculated using Maxwell's equations with certain boundary conditions. The radar cross section of a simple sphere is shown in figure 2 as a function of its circumference measured in wavelengths (2πa/λ), (where a is the radius of the sphere and λ is the wavelength).

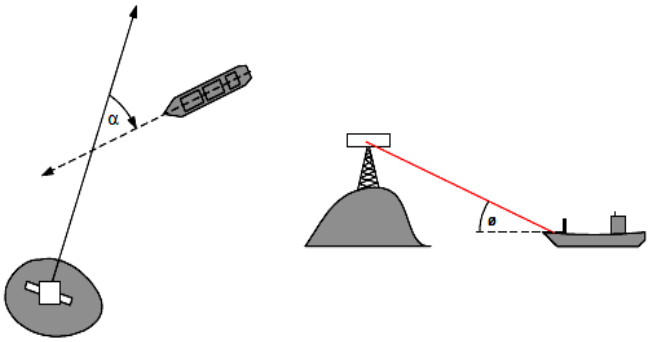


Figure 1: Aspect angle (α) & grazing angle (Ø) of a ship [16].

The region where the size of the sphere is small compared with the wavelength (2πa/λ << 1) is called the Rayleigh region. The RCS of objects within that region varies as λ^{-4}. At the other extreme from the Rayleigh region is the optical region, where the dimensions of the sphere are large compared with the wavelength (2πa/λ >> 1). The RCS approaches the optical cross section πa^2 in the optical region. In between the optical and the Rayleigh region is the Mie, or resonance, region. The cross section is oscillatory with frequency within this region. Since the sphere cross section will not be aspect-sensitive. The cross section of other objects, however, will depend upon the direction as viewed by the radar.

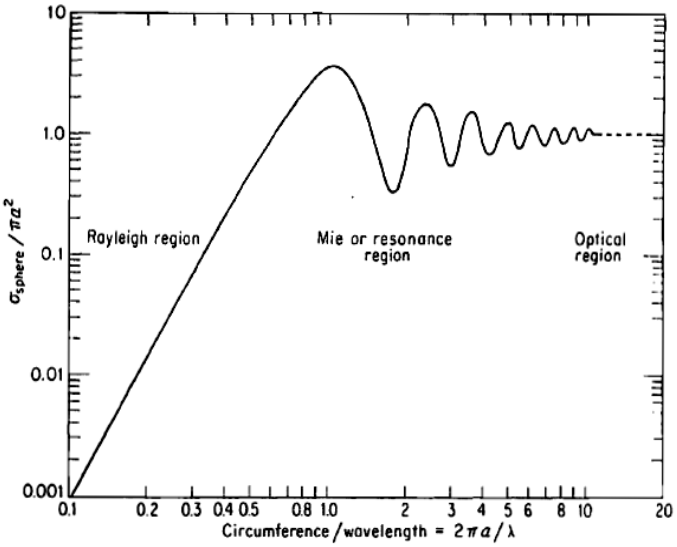


Figure 2: RCS of the sphere. a = radius; λ = wavelength [15].

For *complex objects*, the radar cross section of complex targets such as ships, aircrafts, birds, and terrain can be much more complex than what discussed before, either because of the multiplicity of scatterers on them or because of the complexity of their surface profiles and dielectric constants [17]. A complex target may be considered as comprising a large number of independent objects that scatter energy in all

directions. The energy scattered in the direction of the radar is of prime interest. The relative phases and amplitudes of the echo signals from the individual scattering objects as measured at the radar receiver determine the total cross section. The phases and amplitudes of the individual signals might add to give a large total cross section, or the relationships with one another might result in total cancellation. In general, the behavior is somewhere between total reinforcement and total cancellation. If the separation between the individual scattering objects is large compared with the wavelength-and this is usually true for most radar applications-the phases of the individual signals at the radar receiver will vary as the viewing aspect is changed and cause a scintillating echo. For complex targets there is obviously no firm relation between RCS and radar frequency.

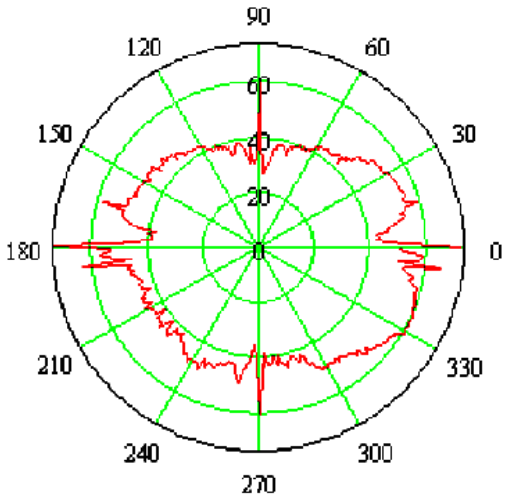


Figure 3: Polar plot of RCS data in Decibels [16].

It is common to present RCS measurements in the form plotted above (fig. 3). This type of graph is called a polar plot, and it is used to chart the RCS data over a range of aspect angles. In this case, the research vessel of 50 m length results have been plotted [16]. The graph of the RCS data (figure 3) shows features, which are quite typical for ships. At bow, stern and broadside aspect angles there are 'highlights' (very strong specular reflections). This polar plot allows us to better see the trends in RCS behavior around the ship. The largest radar returns can be seen from the sides of the ship, where the large surface produces strong reflections. Both the forward and aft aspects also produce large peaks in RCS that are due to bow and stern surfaces. The smallest RCS measurements tend to come from the corners of the aspect envelope where there are no surfaces perpendicular to the radar. It can be difficult to directly compare RCS estimates from one ship to another since the RCS varies wildly depending on aspect angle, radar frequency, and the polarization of the antenna but of course by using NNs, it is easy that pattern recognition from RCS signatures be done quickly so that decisions for dealing with this information can be made in timely manner. In this paper, RCS data are introduced to three models of proposed neural networks which will be discussed in the following section.

3 Proposed Neural Network Structure

Neural networks are composed of many simple elements called neurons working in parallel to solve the classification problems. The aim of NN is to transform the inputs into meaningful outputs. The NN is trained with the available data samples to investigate the relation between inputs and outputs. In this paper, back-propagation based MLP network is used. Back-propagation is the most common algorithm used to train NN because of its ability to generalize well on a variety of problems. Built models will classify patterns or make predictions according to the patterns of inputs and outputs that have been learned. The learning process consists of two phases, feed-forward and back-propagation. During training, an input is presented to the network and propagates to the output layer, then the output is compared with the desired output and the error is back-propagated so that the weights can be adjusted. NN architecture of the type feed-forward back-propagation MLP is used in our approach to classify the RCS signatures of three naval ships of different classes. This network consists of three layers. The first layer is the input layer which accepts input signals from the outside and redistributes these signals to all neurons in the second layer. The input layer does not include any computing neurons. The second layer is the hidden layer. Neurons in the hidden layer detect the features, associated the weights of the neurons in this layer. This features then used by the third layer which is the output layer to determine the output pattern. In this study we proposed three neural network models .*The first model* (NN1) is fed with seventy five inputs corresponding to the data extracted from each RCS polar plot. These inputs are the frequency, the polarization index, seventy two regularly extracted distributed RCS values sampled each 5^o beginning from 5^o and ending at 360^o over one rotation and the mean of these seventy two RCS values, the hidden layer consisted of 60 neurons. The number of neurons in the hidden layer defines by the analysis and it is found that neurons equal to 60 give best results. The number of neurons in the output layer is three for the all three models due to the need to classify the signatures into three classes of ships. *The second model* (NN2) is fed with four inputs which are the frequency, polarization index, the aspect angle and its corresponding RCS value. The hidden layer in this model consisted of 25 neurons. *The third model* (NN3) is fed with six inputs which are the frequency, polarization index, two adjacent aspect angles and their corresponding RCS values. The hidden layer in this model consisted of 35 neurons. The activation function used in the hidden layer and in the output layer for the three models is log-sigmoid transfer function. The results will be classified in the range of zeros to ones. It will indicate the confidence of the results. Figure 4 shows the general structure for the three proposed networks where L = 75 in case of model NN1, L= 4 in case of model NN2 and L = 6 in case of model NN3.

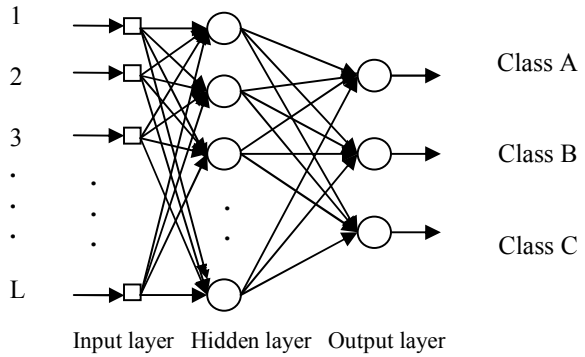


Figure 4: General structure for model NN1, NN2 and NN3.

3.1 Training and testing data

The collected data recorded during RCS measurements are ninety RCS polar plots with known frequency and polarization for three different classes of ships: class A, class B and class C, where 30 polar plots for each class of ships. These data are used as inputs to the NN for training, validation and testing the networks. Using these ninety polar plots with known frequency and polarization, three types of databases were created which are database 1 for model (NN1), database 2 for model (NN2) and database 3 for model (NN3). Table 1, 2 and 3 show the distributions of data base 1, data base 2 and database 3 respectively .Each data base contains three kinds of samples for training, validation and testing each corresponding model.

Table 1: Data base 1 distribution for NN model NN1

Class-category	Number of input patterns	Training data 71.1%	Validation data 14.45%	Testing data 14.45%
Class A	30	Randomly selected 64 input patterns each with 75 dimensions	Randomly selected 13 input patterns each with 75 dimensions	Randomly selected 13 input patterns each with 75 dimensions
Class B	30			
Class C	30			

As shown in table 1: data base 1 contains 90 input patterns each with 75 dimensions which represent the seventy five inputs to model NN1. These inputs are:
1. Seventy two regularly distributed RCS values sampled every 5^o, starting at angle 5^o and ending at angle 360^o are fed to NN1 from input 1 to input 72 and entered in dBsm.
2. The mean value of the last mention 72 RCS values introduced to input 75 and entered in dBsm.
3. The frequency of the coastal monostatic radar entered in GHZ and introduced to input 74.
4. The polarization index where our measurements were done in vertical and horizontal polarizations so we give index (1) to horizontal polarization and index (2) to vertical polarization and introduced this index to input 73.

Table 2: Data base 2 distribution for NN model NN2

Class-category	Number of input patterns	Training data 70%	Validation data 15%	Testing data 15%
Class A	2160	Randomly selected 4536 input patterns each with 4 dimensions	Randomly selected 972 input patterns each with 4 dimensions	Randomly selected 972 input patterns each with 4 dimensions
Class B	2160			
Class C	2160			

As shown in table 2: data base 2 contains 6480 input patterns resultant from multiplied 90 polar plots by 72 sampling points in each polar plot .Each input pattern has four dimensions which represent the inputs to model NN2. These inputs are:
1. The polarization index in which our measurements were done in vertical and horizontal polarizations so we give index (1) to horizontal polarization and index (2) to vertical polarization and introduced this index to input 1.
2. The frequency of the coastal monostatic radar entered in GHZ and introduced to input 2.
3. The aspect angle entered in degree & introduced to input 3.
4. The corresponding RCS value to the pervious input aspect angle entered in dBsm and introduced to input 4.

Table 3: Data base 3 distribution for NN model NN3

Class-category	Number of input patterns	Training data 70%	Validation data 15%	Testing data 15%
Class A	4320	Randomly selected 9072 input patterns each with 6 dimensions	Randomly selected 1944 input patterns each with 6 dimensions	Randomly selected 1944 input patterns each with 6 dimensions
Class B	4320			
Class C	4320			

As shown in table 3: data base 3 contains 12960 input patterns resultant from multiplied 90 polar plots by 72 sampling points in each polar plot then repeating the resultant two times due to the effect of adding the adjacent aspect angle which equals the aspect angle $\pm 5^o$. Each input pattern has six dimensions which represent the inputs to model NN3. These inputs are:
1. The polarization index in which our measurements were done in vertical and horizontal polarizations so we give index (1) to horizontal polarization and index (2) to vertical polarization and introduced this index to input 1.
2. The frequency of the coastal monostatic radar entered in GHZ and introduced to input 2.
3. The aspect angle entered in degree & introduced to input 3.
4. The corresponding RCS value to the pervious input aspect angle entered in dBsm and introduced to input 4.
5. The adjacent aspect angle in degree which equals to the aspect angle $\pm 5^o$ and introduced to input 5.
6. The corresponding RCS value to the pervious input adjacent aspect angle entered in dBsm & presented to input 6.

3.2 Training stage

During training, the weights and biases of the network are iteratively adjusted to minimize the network performance function in each model. The default performance function for feed-forward networks is the mean square errors which equal the average squared errors between the network outputs and the target outputs. In the training process, the training data are fed into the input layer. The input features then are propagated to the hidden layer and then to the output layer. This is called the forward pass of the back-propagation algorithm. The output values of the output layer are compared with the target output values. If any value is different, an error is calculated and then propagated back toward hidden layer. This is called the backward pass of the back-propagation algorithm. The error is used to update the connection strengths between neurons. The network was trained by using Scaled Conjugate Gradient (SCG) back-propagation algorithm. SCG is fully automated; independent parameters and avoids a time consuming line search [18].

3.3 Testing stage

After creating the networks, the 13, 972 and 1944 new data samples were used for testing the networks for model NN1, NN2 and NN3 respectively. During testing phase, no learning takes place so weights are not changed. The converged weights that are obtained during the training process are then loaded into the networks. The outputs are obtained in a feed-forward method. In order to predict success of classifier, the classification accuracy was calculated by comparing the actual outputs with the classified one.

4 Results and Discussion

In this study we used the Matlab which includes the neural network toolbox [19]. This software is used to simulate our three proposed models and then to train these three models and to illustrate the classification results through confusion matrices.

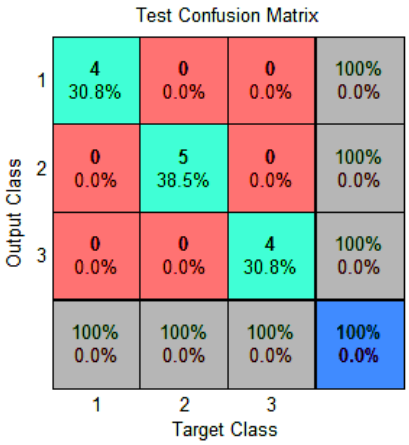


Figure 5: Test confusion matrix for NN1

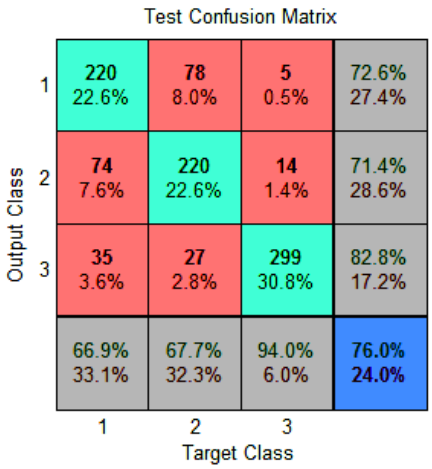


Figure 6: Test confusion matrix for NN2.

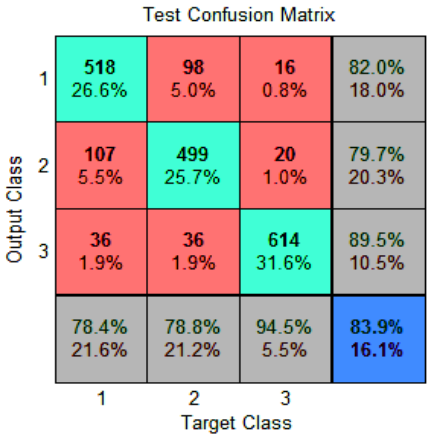


Figure 7: Test confusion matrix for NN3.

After the training phase, testing the Scaled Conjugate Gradient (SCG) back-propagation algorithm was established. As mentioned earlier, three neural network models are used to classify the ship RCS signals which are NN1, NN2 and NN3. For the first model (NN1), seventy five inputs were presented to the network structure. After the trained network was tested by the data for testing process, the group of class A was classified correctly with 100%. The group of class B was classified correctly with 100%. Then, the group of class C was classified correctly with 100% where the results are shown in confusion matrix form in figure 5. For the second neural network model (NN2), four inputs were presented to the network structure. After the trained network was tested by the data for testing process, the group of class A was classified correctly with 66.9 % .The group of class B was classified correctly with 67.7%. Then the group of class C was classified correctly with 94 % so the overall correct classification was 76% for the three classes A, B and C as shown in figure 6.

Table 4: The classification accuracy for each class of ships (A, B & C) for network structures NN1, NN2 and NN3

Classification results									
	NN1			NN2			NN3		
Class type	No. of testing samples	True	False	No. of testing samples	True	False	No. of testing samples	True	False
Class A	4	4	0	329	220	109	661	518	143
Class B	5	5	0	325	220	105	633	499	134
Class C	4	4	0	318	299	19	650	614	36
Total no.	13	13	0	972	739	233	1944	1631	313
Total percentage	…	100%	0%	…	76%	24%	…	83.9%	16.1%

For the third neural network model (NN3), six inputs were introduced to the network structure. After the trained network was tested by the data for testing process, the group of class A was classified correctly with 78.4%. The group of class B was classified correctly with 78.8 %. Then the group of class C was classified correctly with 94.5% so the overall correct classification reaches 83.9% as shown in figure 7. From the previous results of the test confusion matrices for models NN1,NN2 and NN3 showed in figures 5,6 and 7 we can notice that for model NN1 there is no misclassification because we make the classification according to the whole RCS signature of the target at certain frequency and polarization but this requires to know the RCS values during one rotation of the target which may be not easy in actual scenario because the target may not make a full revolution during the observation time, for that reason we decided to construct the second model NN2 where for the class A there were 220 correct classifications , 74 samples were misclassified as class B instead of class A and 35 samples were misclassified as class C in instead of class A. For the class B there were 220 correct classifications, 78 samples were misclassified as class A instead of class B and 27 samples were misclassified as class C in instead of class B. For the class C there were 299 correct classifications, 5 samples were misclassified as class A instead of class C and 14 samples were misclassified as class B in instead of class C. Generally the model NN2 has lower correct classification than model NN1 because in model NN2 classification is according to only one RCS value at certain aspect angle and certain frequency and polarization also misclassification rate increases between class A and class B in model NN2 due to the similarity in size between the two classes of ships. We increased the overall correct classification achieved in model NN2 from 76% to about 84% in model NN3 by using two RCS values at two adjacent aspect angles at certain frequency and polarization instead of one RCS value at one aspect angle at certain frequency and polarization. Table 4 is shown above gives us a conclusion of the classification results while figure 8 illustrates the correct classifications of class A, B and C for network structures NN1 , NN2 and NN3.

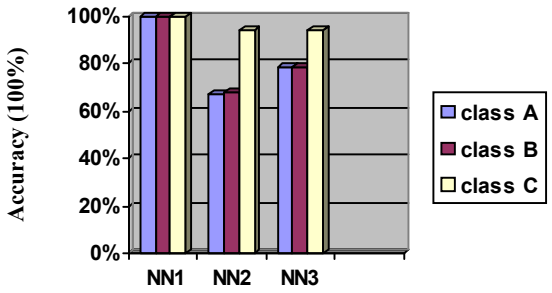


Figure 8: Classification accuracy of class A, class B and class C for NN1, NN2 and NN3

5 Conclusions

In this paper, three neural network-based classification schemes are used to classify three different types of real ships according to their RCS signatures by the aiding of collected data from RCS measurements. These measurements were done by coastal monostatic radar in the frequency range (2-18GHZ) with horizontal and vertical polarizations at low grazing angle and at good weather and sea state conditions. Three different classification models are used with this implementation. In the first model, classification according to 75 inputs extracted from the whole RCS polar plot and presented to model NN1. The classification system for NN1 results in 100% correct classification. The second model gives us a rapid classification decision because it classifies target ships according to only one RCS value at certain aspect angle and certain frequency and polarization where 4 inputs are introduced to model NN2. This model results in a correct classification 76%. In order to increase the correct classification in the second model, the third model was established to classify target ships according to two RCS values at two adjacent aspect angles at certain frequency and polarization where 6 inputs are presented to model NN3. The classification system reaches 84% correct classification. The study indicates that the NN based classification scheme provides excellent performance in targets classification, especially when 72 equally distributed RCS values from the whole RCS polar plot are used by model NN1 for classification. Also provides good performance with

very fast and practical simple method when two RCS values at two adjacent aspect angles are used by model NN3. We have shown that it is possible to obtain very good classification of RCS signatures, with no preprocessing. This reduction in computational load can have significant impact in some applications because radar may be the only system that capable of separating enemy targets from friendly ones in timely manner.

6 Future Work

1. Consider the factors of weather and sea state conditions in the inputs to the neural networks by collecting different RCS measurements at different weather and sea state conditions.
2. Use another artificial intelligence tool such as fuzzy logic for the classification purpose and compare with our obtained results.

7 References

[1] N. Sandirasegaram and C. Liu, "Analysis of High Resolution Polarimetry Data of Static Targets in Automatic Target Recognition Context" DRDC Ottawa TM 2007-330 ; Defence R & D Canada – Ottawa; December 2007.

[2] L. F. Araghi, H. Khaloozade and M. R. Arvan, "Ship Identification Using Probabilistic Neural Networks (PNN)", Proceedings of the International Multi Conference of Engineers and Computer Scientists 2009 Vol. II IMECS 2009, March 18 - 20, 2009, Hong Kong.

[3] Philip D. Heermann and Nahid Khazenie, Member, IEEE, "Classification of Multispectral Remote Sensing Data Using a Back-Propagation Neural Network." ,IEEE Transactions on Geoscience and Remote Sensing. Vol. 30, No. 1, January 92.

[4] M. Ivanova," An Application of Self-Organizing Maps in the Process of Classification. ", M.Sc. thesis, Faculte des Sciences et De Genie Universite Laval Quebec 2008.

[5] A. Elgerzawy, "Naval Targets Identification Using Passive Sonar Signals" , thesis presented to obtain the degree of M.Sc., Electrical Engineering Department, Alexandria University, Faculty of Engineering, 2008.

[6] H. Hahem, "Underwater Targets Recognition" , thesis submitted to the Electrical Engineering Department for the degree of Doctor of Philosophy, Alexandria University, Faculty of Engineering, 1998.

[7] N.K. Ibrahim, R.S.A. Raja Abdullah and M.I. Saripan, "Artificial Neural Network Approach in Radar Target Classification", Journal of Computer Science 5 (1): 23-32, 2009. ISSN 1549-3636. © 2009 Science Publications

[8] R. Soleti, L. Cantini, F. Berizzi, A. Capria and D. Calugi, "Neural Network for Polaremtric Radar Target Classification", 14th European Signal Processing Conference 2006, Florence, Italy, September 4-8, 2006, copyright by Eurasip.

[9] S. Chakrabarti, N. Bindal, and K. Theagharajan, "Robust Radar Target Classifier Using Artificial Neural Networks", IEEE Transactions on Neural Networks, vol. 6, No. 3, May 95.

[10] I. Jouny, Member, IEEE, E D. Garber, Member, IEEE and S. C. Ahalt, Member, IEEE, "Classification of Radar Targets Using Synthetic Neural Networks" ,IEEE Transactions on Aerospace and Electronic Systems Vol. 29, NO.2 April 93.

[11] J. Wenli, Z. Huoju, L. Qizhong and Z. Yiyu, "Efficient Radar Target Classification Using Modular Neural Networks", National University of Defense Technology, Changsha, China, 2001 IEEE.

[12] A. Hossen, F. Al-Wadahi and J. Jervase," Classification of Modulation Signals Using Statistical Signal Characterization and Artificial Neural Networks", 2006 Elsevier Ltd, published in Journal Engineering Applications of Artificial Intelligence, vol.20 issue No. 4 June, 2007 available online at www.sciencedirect.com

[13] M. F. Byl, J. T. Demers and E. A. Rietman ,"Using a Kernel Adatron for Object Classification with RCS Data", Approved for public release Storming Media LLC, arXiv:1005.5337v1 [cs.LG] 28 May 2010

[14] G. P. Zhang, " Neural Networks for Classification: A Survey", IEEE Transactions on Systems, Man, and Cybernetics —PART C: Applications and Reviews, Vol. 30, No. 4, November 2000.

[15] Merrill I. Skolnik, "Introduction to Radar Systems", Second Edition, McGraw-Hill Book Company.

[16] I. Harre," RCS in Radar Range Calculations for Maritime Targets" available online at http://www.mar-it.de/Radar/RCS/RCS_xx.pdf

[17] Merrill I. Skolnik, "Radar Handbook", Third Edition, Copyright © 2008 by the McGraw-Hill Companies.

[18] M. Engin, S. Demirag, E.Z. Engin, F. Ersan, E. Asena and Zafer C., "The classification of human tremor signals using artificial neural network" , 2006 Elsevier Ltd, available online at www.sciencedirect.com

[19] Mark H. B., M. T. Hagan and H. B. Demuth, " Neural Network Toolbox™ 7", User's Guide, Copyright 1992–2010 by the MathWorks, Inc. and available online at www.mathworks.com

Multi-Layer Perceptrons and Conventional Adaptive Filters for Channel Estimation in CDMA System

Sahar Nasrzadeh[1], Mahdih Gehasemlou[1], Mehrdad Jalali[2]
[1]Telecommunication Department, Islamic Azad University, South Tehran Branch, Iran
[2]Artificial Intelligent Department, Islamic Azad University, Mashhad Branch, Iran

Abstract— *In this paper, two different approaches multi-layer perceptron (MLP) neural network (NN) with back propagation (BP) learning algorithm and conventional adaptive filters are tested and compared in AWGN channel for code division multiple accesses (CDMA). In this work direct-sequence (DS) CDMA with binary phase shift keying (BPSK) modulation transmitted over complex base-band channel for real part of the matched-filter (MF) output and not the MF output itself should be use as sufficient statistics for further processing. Based on this observation, we drive novel improved bit error rate (BER) and mean square error (MSE) in MLP NN receiver. A pseudo noise (PN) has produced by stochastic statistical properties and PN is changed in duration of each symbol. Results show that MLP-NN has better performance than other proposed MFs in which training sequence is transmitted at the beginning of a session continuously and the initial estimation of the channel parameters is performed using the received data symbols.*

Keywords: Multi layer perceptrons (MLP), Neural network (NN), Back propagation (BP), Kalman filter (KF), Code division multiple accesses (CDMA).

1 Introduction

The performance of non-orthogonal DS-CDMA systems can be significantly improved by multi-user detection [5]. The optimum multi-user detector was proposed by verdu [6]. The high complexity of the optimum receiver has been the motivation for design of numerous suboptimum multi-user receivers.

With the growing interest in the application of adaptive filters to signal processing, there is a more urgent need to understand and compare NN- based algorithms with more conventional approaches in terms of theory, performance practical engineering issues.

Artifical NN (ANN) based on their basic properties of universal approximation and learning capabilities have properties that enable their application in data-based process modeling. In addition, NN is applicable for channel estimation because of its strong approximation and learning ability. In this paper, a kind of NN, MLP is applied to CDMA for channel estimation. Within the whole study the MLP algorithm is preferred over the general ANN algorithms. This static NN is also simple in structure, stable and easier to train as compared to dynamic NN [7].The BP is a training algorithm to MLP that standard BP is a steepest descent algorithm[8]. The number of hidden layer (HL) that is consisted of computation nodes provides performance of approximation. In this research single HL is proposed for MLP. Since increasing the number of layer, becomes rising complexity of the system and heavier models, instead of using more layers can be increased neurons numbers of layer, That larger the number of hidden nodes cause more accurate the approximation.

All presented receivers were originally designed for DS-CDMA with BPSK modulation over AWGN multipath fading and single channel. The filters of the later receivers can be conveniently adjusted to use adaptive algorithm and estimate channel parameters.

The analysis of the BER for DS-SS assumes that K-1 is the interfered spread sequences, random and N is chips long. The BER is obtained by averaging over all possible spreading sequences, including the desired sequence.

The rest of paper is organized as follows. In Section 2 and 3, concept of CDMA and the system model are presented. In Section 4, MLP neural network with BP learning algorithm is presented for channel estimation. In Section 5, the Kalman filter is presented. the simulation results are given in Section 6. In Section 7 makes some conclusions.

This work was funded in part by Ghiyasodin Jamshid Kashani Institute, Tehran, Iran.

2 Learning algorithm for CDMA signals

CDMA, also known as spread-spectrum multiple accesses (SSMA), provides a means of separating the signals of multiple users transmitting simultaneously and occupying the same RF bandwidths. Specifically, there is no hard limit on how many users can simultaneously share the channel [3].
Each user's waveform is like an independent noise random process .In a DS CDMA system, each user has a distinct PN code (or sequence) which their cross-correlation is nearly zero. The message from each user is modulated with corresponding PN code, resulting in a transmission bandwidth much greater than the message bandwidth. Moreover, there is an integer number of chips for each data symbol for the stochastic chips with the width T_c. Binary noise sequence is mapped to a chip spreading sequence of +/-1's.
The receiver correlate to the code of the desired user, to receive the signal of user 1, we should correlate the received signal with user 1's spreading sequence. Each undesired user' code has small amount of residue.
The performance of non-orthogonal direct DS-CDMA system can be significantly improved by multiuser detection [2]. MF pulls out desired user' waveform and suppresses the interference.
One of many reasons for spreading the spectrum is the inherent immunity of the communication system to interference.
Due to the spreading properties, the channels of code-division are semi-orthogonal. It's an important property that these codes allow to each cell to reuse the channel, but decreasing the interference of all users within the cell (intracellular interference) is similar to the other users in the other cells (intercellular interference).

3 A model for the received signal

Assuming a synchronous DS/CDMA system with M users. The ith bit transmitted by the mth user is $b_{m,i}$ and the signature waveform of the mth user $P_m(n)$, which is restricted to a bit interval of duration T and $T = CRT_c$, assumed to be orthogonal, thus

$$\sum_{n=0}^{T} \left| P_m(n) \right|^2 = 1 \tag{1}$$

Consider a narrowband signal model is assumed. Received signal in base band in multipath fading AWGN channel is expressed:

$$r(n) = \sum_{l=1}^{L} \left[\sum_{i=1}^{T} \sum_{m=1}^{M} b_{m,i} P_m(n - iT) u_{ml} + n_l(n) \right] + s(n) \tag{2}$$

For single user L=1.
In this paper, for simplicity of simulation, L was assumed equal to M (user number).
Where u_{ml} is the array response vector of the mth user in the path, $n_l(n)$ is the vector of the additive complex Gaussian noise in the lth path and it is especially white. $s(n)$ is additive channel noise with variance $\Phi^2 = N_0 / 2$.

$$E\left[n(n) n^*(n-\tau) \right] = \sigma^2 \delta(\tau) \tag{3}$$

$$E\left[s(n) s^*(n-\tau) \right] = \Phi^2 \delta(\tau) \tag{4}$$

Where δ is kronecker delta function.

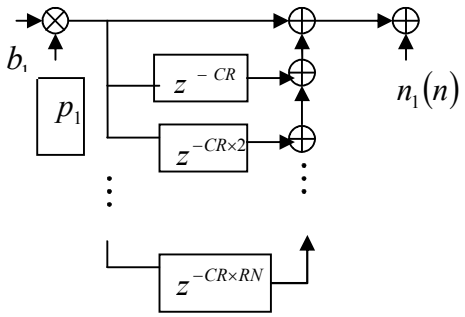


Fig.1 An entire diagram model of the received signal.

Where RN is the Reputation Number and its value is the range of the one to RN.

4 MLP network

ANN's because of their inherently nonlinear nature are well suited to signal processing application. The general goal for ANN signal processing algorithms in practical applications is given noisy and imprecise nonlinear data, to enhance desired responses and reduce irrelevant and unwanted responses. ANN's can be classified into two broad class: supervised and unsupervised learning models. The unsupervised or self-organizing models group the input samples into self-similar classes based on some specific measure of similarity. The supervised learning models are trained by exposing them to example input/output

vector pairs to implement mappings the match the examples or underlying equations as closely as possible. Notable example of these is the MLP NN. The MLP NN has been the most popular despite its deficiencies with respect to network design and learning and adaption complexity.

MLPs are feed-forward networks with one or more hidden layers. The NN employed in this study as can be seen in figure 2, possessed a three-layer learning network consisting of an input layer, a hidden layer and an output layer. The most common learning rule for MLPs is the BP algorithm (BPA). BP is a supervised learning algorithm and involves two passes: a forward pass in which input vector is applied to input nodes then its effects propagate through the network layer-by-layer with fixed synaptic weights, and a backward pass in which synaptic weights are adjusted in accordance with error signal that are made differences between the computed and observed information signals at the output units. In this study BPA is used to training MLPs. Tangent sigmoid function is used as neuron transfer function that is a non-linear function. Nonlinearity behavior of MLP is caused different applications to a broad class of nonlinear signal processing problems that can be realized with present or foreseeable hardware technology. The activation function of the hidden layer is

$$net_j = \sum_{j=1}^{d} X_i w_{ij} \quad (5)$$

$$o_j = f(net_j) \quad (6)$$

Where d is number units, X_i is input data to the network and w_{ij} is input-to-hidden layer weights at the hidden node j when apply activation function. Each output nodes similarly computes [9]. In training process, weights are computed by minimizing

$$E_{av} = \frac{1}{N} \sum_{n=1}^{N} E(n) \quad (7)$$

,

$$E(n) = \frac{1}{2} \sum_{j \in C}^{N} e_j^2(n) \quad (8)$$

Where C is set of output nodes , $e_j(n) = d_j(n) - y_j(n)$, $d_j(n)$ and $y_j(n)$ are jth desired output and jth network output. The training process is stopped when the value of E_{av} of the goal is given.

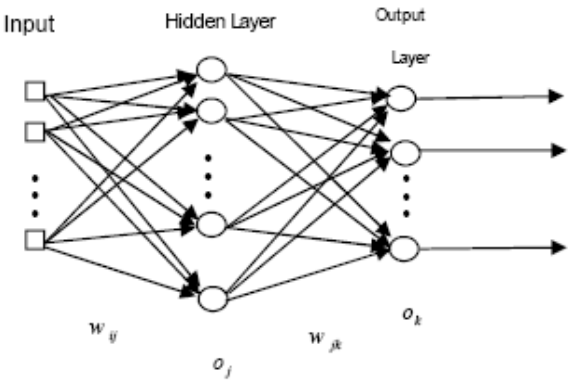


Fig. 2. The MLP structure

Also according to universal approximation theorem, a single hidden layer is sufficient for MLP to compute a uniform error approximation to a given training set.

5 The Kalman filter

In this paper, the KF structure is as same as algorithm that presented in [4], [1].The Kalman estimator is the optimal solution to the continuous or discrete problems. The full text explanations of equations were discussed in [2].

6 Computer simulation results

In this section, the performance of three detectors MLP-NN, KF and RLS was compared in single AWGN channel while synchronization was accomplished in receiver. Since the system is chip-synchronous, the received signal is sampled at chip rate at output of a chip-matched filter. Spreading codes was chosen stochastic entirely because this is an essential condition to provide orthogonally in spread spectrum system.

We assume that the receiver perfectly knows the user's spreading sequence.

The simple structure of MLP NN is chosen:1 hidden layer, a sequence of 3000 bit ,3000 neurons that is computed from geometry mean in the hidden layer, activation functions: 'tansig' in the hidden layer and ' purelin' in the output layer. The Levenberg-Marquardt (LM) algorithm is used for training. All the results are obtained after repeatedly training the NN.

Fig.3 shows the BER curves versus signal to noise ratio (SNR) for the five users with 3000 bits data for each user. A spreading sequence length of CR=8 was used. The suggested detector NN has better performance as compared with other detectors and this superior continued to ten dB while it was equal to zero for SNR=12dB and SNR=13dB.

Fig. 4 depicts the efficiency of suggested NN and KF estimator in AWGN channel according to mean squared error (MSE) in case of five users in CDMA users having the 6 process gain. The number of chips per symbol is the processing gain (PG). This also $PG = \frac{B_{SS}}{B}$ where B_{SS} and B are the bandwidth of the chips and the data symbols, respectively.
Usually, $B_{ss} \rangle\rangle B$. It mean that after 250th received bit to the end real out put is equal to estimated output by Kalman filter, while estimated output by MLP-NN started with low MSE and it was stable along total sequence data.

Fig. 5 shows a typical output of an estimator based on MLP with 1000 hidden neurons. Result is shown after training and testing on the data obtained from AWGN channel. This result describes the NN output after training is similar to channel output.

7 Conclusions

In this paper, adaptive filters were considered for the symbol detection and channel estimation. The idea for using these filters is that the signal of interest was destroyed by MAI and interference from adjacent cell which causes to add high power noise in each path when they are independent from each other.

The high complexity calculation of these filters in multi path fading AWGN channel justified or proved the better performance of these filters with low probability of error when we use new structure of CDMA signal training sequence which was described.

A channel estimation method based on MLP NN with BPA is proposed in this paper. Determining an appropriate architecture of a NN for a particular problem is an important issue since the network topology directly affects its computational complexity and its generalization capability. Simulation results show that the MLP has better performance than conventional adaptive filter.

The NN cell is a processor and each of its connections has special weight coefficients. From simulation result it's clear that NN input-output mapping is built by learning from training sequences and the other advantage is adaptively that causes to gain adapt synaptic weight to changes of environment. Because of previous advantages, NN can apply to adaptive signal processing.

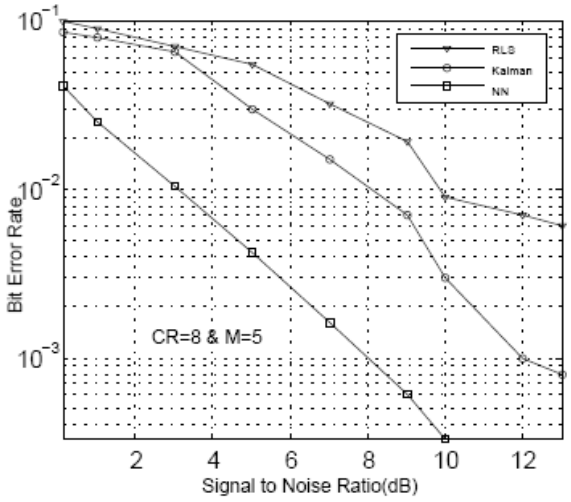


Fig. 3. Bit Error Probability versus SNR in AWGN channel

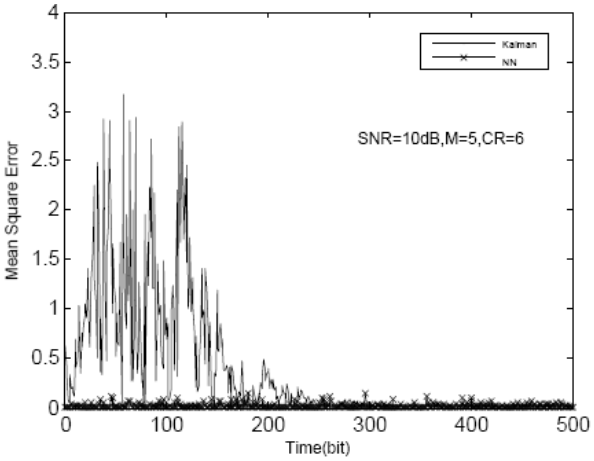


Fig. 4. Conversion of NN and Kalman curves in AWGN channel

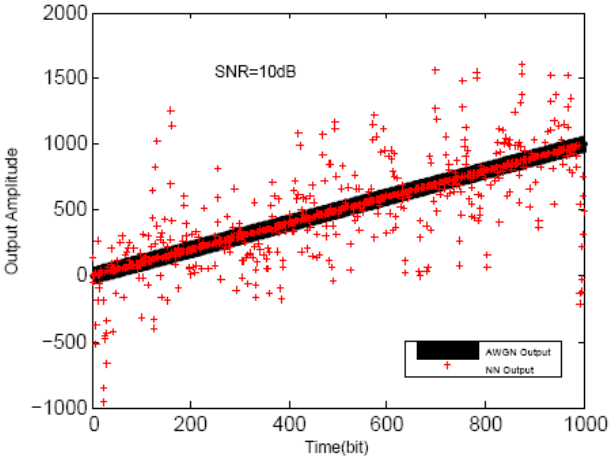


Fig. 5. Output of estimator based on MLP in AWGN channel

8 References

[1] T. J. Lim, L. K. Rasmussen, and H. Sugimoto. "An asynchronous multiuser CDMA detector based on the kalman filter"; IEEE J.Select.Areas Commune., vol.16, No. 9, pp.1711-1722, Dec.1998.

[2] G. Welch and G. Bishop. "An introduction to Kalman filters"; Department of computer Science University of north Carodina at Chapter Hill, July 2006

[3] Jan SIMSA. " Simulation of Blind Adaptive DS-CDMA LMS Detector ," 978-1-4244-3538-8/09/$25.00 2009 IEEE.

[4] F. Gustafson. "Adaptive Filtering and change detection"; J. Wiley & sons, England, 2000.

[5] S. Verdu. "Multiuser detection", Cambridge University Press, New York, 1998.

[6] S. Verdu. "Minimum probability of error for asynchronous Gaussian multiple-access channels", IEEE Transe. Inform. Theory IT-32 , pp. 85-96, January 1986.

[7] Drazen Siskovic. " Estimation of Difficult-to-Measure Process Variables Using Neural Network";IEEE MELECON, May12-15, 2004.

[8] Mohamed M.A. Moustafa , Salwa H.A. El-Raml., "Channel Estimation and Equalization Using Back Propagation Neural Networks in OFDM Systems"; IEEE 2009.

[9] Necmi Taspinar, M. Nuri Seyman. " Back Propagation Neural Network Approach for Channel Estimation in OFDM System"; IEEE 2010.

Neural Network Based Approach for Automotive Brake Light Parameter Estimation

A. V. Ortega and I. N. da Silva
Department of Electrical Engineering, University of São Paulo, CEP 13566-590, São Carlos, SP, Brazil

Abstract - *The advantages offered by the electronic component LED (Light Emitting Diode) have caused a quick and wide application of this device in replacement of incandescent lights. However, in its combined application, the relationship between the design variables and the desired effect or result is very complex and it becomes difficult to model by conventional techniques. This work consists of the development of a technique, through artificial neural networks, to make possible to obtain the luminous intensity values of brake lights using SMD (Surface Mounted Device) LEDs from design data. Such technique can be used to design any automotive device that uses groups of SMD LEDs. Results of industrial applications, using SMD LED, are presented to validate the proposed technique.*

Keywords: Brake light, SMD LED, neural networks, intelligent systems.

1 Introduction

The LED device is an electronic semiconductor component that emits light. At present time, it has been used in replacement of incandescent lights because of its advantages, such as longer useful life (around 100,000 hours), larger mechanic resistance to vibrations, lesser heating, lower electric current consumption and high fidelity of emitted light color [1].

However, in designs where incandescent lights are replaced by LEDs, some of their important characteristics must be considered, such as direct current, reverse current, vision angle and luminous intensity [2].

The SMD LEDs are suitable for use in a wide variety of electronic equipment, including cordless and cellular phones, notebook computers, hand-held products, network systems, and automotive interior applications.

In automobile industry, incandescent lights have been replaced by LEDs in the brake lights, which are a third light of brakes [3]. In these brake lights are used sets of SMD LEDs usually organized in a straight line. The approval of brake light prototypes is made through measurements of luminous intensity in different angles, and the minimum value of luminous intensity for each angle is defined according to the application [4]. Several aspects related to the physical properties of luminous intensity can be found in [13].

The main difficulty found in the development of brake lights is in finding the existent relationship between the following parameters: luminous intensity (I_V) of the SMD LED, distance between SMD LEDs (d) and number of SMD LEDs (n), with the desired effect or result, i.e., there is a complexity in making a model by conventional techniques of modeling, which are capable to identify properly the relationship between such variables. The prototype designs of brake lights have been made through trials and errors, causing increasing costs of implementation due to time spent in this stage. Moreover, the prototype approved from this system cannot represent the best relationship cost/benefit, since few variations are obtained from configurations of approved prototypes. The artificial neural networks are applied in cases like this one, where the traditional mathematic modeling becomes complex due to nonlinear characteristic of the system. These networks are able to learn from their environment and to generalize solutions, making them attractive to this type of application.

More specifically, multilayer perceptron artificial neural networks are used to estimate all values of luminous intensity required in brake light designs, which use SMD LEDs in their structures. In these cases, several brake light configurations can be simulated from the proposed approach, and those future prototype configurations that will meet the minimum values of luminous intensity required by traffic (vehicle safety standards) regulations can be identified.

2 Overview of automotive applications using LEDs

Modern automotive vehicles use incandescent lamps for parking, turning, and brake lights. These red and yellow lights typically employ a standard clear incandescent bulb behind a colored lens. However, incandescent bulbs consume a disproportionately large amount of energy for the amount of colored light they project from the vehicle's lighting fixture.

Recently, automotive industries have supported the development of schemes that replace the inefficient incandescent lights described above with more efficient lights like the Light Emitting Diode. Because LEDs produce light at the wavelength necessary for automotive use, less energy is consumed by these lighting fixtures than those that use white

light generated by incandescent bulbs. Moreover, LEDs exhibit long lifetimes that are on the order of 100,000 hours. Coupled with the ruggedness inherent in solid-state devices, this indicates that LEDs may be useful for low maintenance applications. Fast response times also make them ideal for some automotive equipment.

In [4] is demonstrated that the conversion of a turn signal from an incandescent light to LED is possible with the latest advancements in LED designs. In [1] is proposed a system based on LEDs for vehicle traffic control applications. From geometric considerations, the system requires a cluster of 200 red, amber, and green or 200 multicolor LEDs for a single three-light system. In [3] is presented a vehicle that uses LEDs in its headlights. To obtain white light from an LED, a blue LED was placed behind a phosphor that emits yellow light when stimulated by the blue. Since yellow light stimulates the red and green receptors of the eye, the resulting mix of blue and yellow light gives the appearance of white (often called “lunar white”).

In [8] is described a light-emitting diode brake-light messaging (LEDBM) system that can be used to avoid rear collisions. The LEDBM is comprised of modulated LED brake lights that communicate information about a vehicle's state to any following vehicle that is equipped with an LEDBM receiver. In [11] is proposed a robust vehicle detection method that uses vision to extract bright regions brake lights. In [12] is presented a vision system dedicated to the detection of vehicles in reduced visibility conditions; this system can identify brake light luminosity in order to avoid collisions.

This paper presents an industrial application using artificial neural networks to estimate values of brake light luminous intensity from design data. Although this study is aimed at the application of LED in brake lights, the methods developed and described here can also be used in other applications, such as headlights, turn lights, rear lights, traffic lights, or any other application where SMD LEDs can be used in groups.

3 SMD LEDs applied in brake lights

LED is an electronic device composed by a chip of semiconductor junction that when traversed by an electric current provides a recombination of electrons and holes. Figure 1 shows the representation of a junction being polarized.

However, this recombination demands that the energy produced by free electrons can be transferred to another state. In semiconductor junctions, this energy is released in form of heat and by emission of photons, i.e., light emission [5]. In silicon and germanium the largest energy emission occurs in form of heat, with insignificant light emission. However, in other materials, such as GaAsP or GaP, the number of light photons emitted is sufficient to build a source of quite visible light [6]. This process of light emission, which is intrinsic characteristic of the LEDs, is called electroluminescence [7]. In Fig. 2 can be observed the representation of the basic structure of a SMD LED.

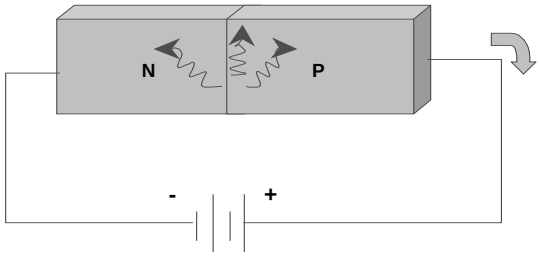


Fig. 1. Junction PN being polarized.

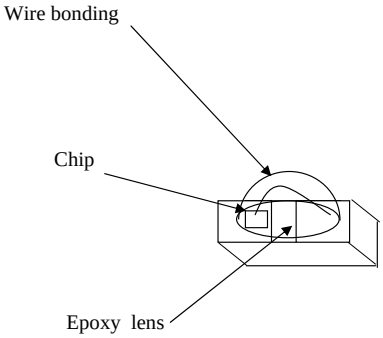


Fig. 2. Basic representation of SMD LED structure.

In brake lights the SMD LEDs are applied in set and generally organized in a straight line on a printed circuit board (PCB). In this PCB, besides the SMD LEDs, there are electronic components, basically resistors, which are responsible for the limitation of electric current that circulates through the SMD LEDs.

The main parameters used in brake lights designs are given by: SMD LED luminous intensity (I_V), distance between SMD LEDs (d) and number of SMD LEDs (n). In Fig. 3 is illustrated a basic representation of a brake light.

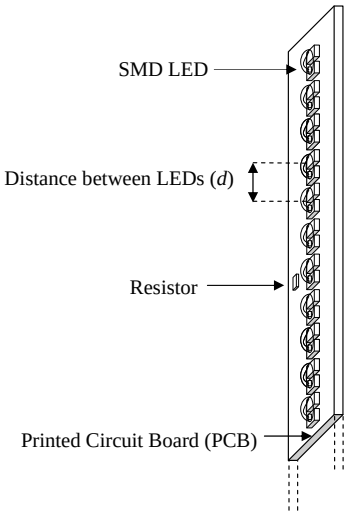


Fig. 3. Representation of a brake light.

The main function of the brake light is to increase the safety of the vehicle (acting as a prevention system) and to reduce the risk of back collisions. Recent studies show the development of brake lights equipped with modulated signal transmitters containing information about the vehicle in which it is installed. Other vehicles that have the respective reception system of those modulated signals receive them, and their information's have been used to prevent back collisions [8]. In Fig. 4 is illustrated a brake light installed.

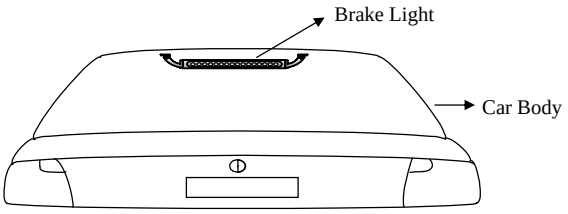


Fig. 4. Representation of a brake light installed.

At the moment there is no model or technique for designing brake lights and the prototypes are elaborated according to the common sense of designers, i.e., through trial and error methods. This occurs because the relationship between the variables involved with the light emission process of brake lights is completely nonlinear.

After elaboration of the brake light prototype, it is necessary an approval of the sample. The process for the prototype validation is made by measuring the luminous intensity of the brake light in 18 positions or different angles (Fig. 5). After this process, the values obtained in each angle are compared with those values established by governmental rules. The minimum value of luminous intensity (I_{VBL}) in each angle varies according to the application. In Figure 5 is shown a representation of a generic distribution diagram of brake light luminous intensity (I_{VBL}) in relation to angle. The mean horizontal position is indicated by 0^oH and the mean vertical position is indicated by 0^oV. Thus, the position defined by the pair of angles (0^oV, 5^oL) is represented by the shaded position shown in Fig. 5.

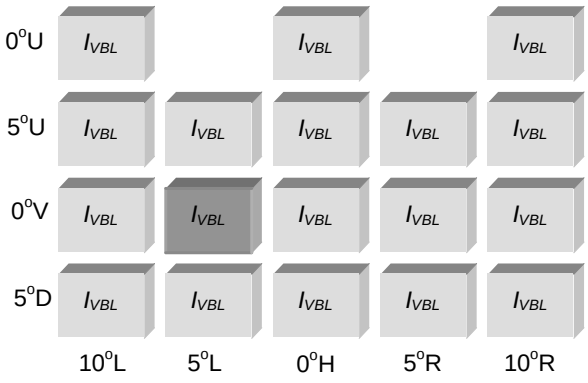


Fig. 5. Generic diagram of luminous intensity (I_{VBL}) in relation to angle.

4 Materials and methods

For this study, 45 samples of brake lights were constructed with the following parameter variations:

- Distance between SMD LEDs (d): 4.5 mm, 5.5 mm and 6.5 mm.
- Number of SMD LEDs (n): 16, 22 and 28.
- Luminous intensity of SMD LED (I_v): 600 mcd, 800 mcd, 1200 mcd, 1500 mcd and 1800 mcd.

This combination of parameters referring to each sample can be seen in Table I. It is important to remember that the minimum and maximum values of each parameter in the designed samples must be chosen in such a way as to represent the domain for parameter variation in future designs, because these designs will be made using the proposed neural network.

A photometer was used to measure the luminous intensity of the samples, and it was coupled to a device permitting vertical and horizontal angle variation. In this way, it was possible to obtain the luminous intensity value from 18 different angles.

Table I. Combination of parameters in each sample.

Sample	d (mm)	n (unit)	I_V (mcd)	Sample	d (mm)	n (unit)	I_V (mcd)
01	4.5	16	600	24	5.5	22	1500
02	4.5	16	800	25	5.5	22	1800
03	4.5	16	1200	26	5.5	28	600
04	4.5	16	1500	27	5.5	28	800
05	4.5	16	1800	28	5.5	28	1200
06	4.5	22	600	29	5.5	28	1500
07	4.5	22	800	30	5.5	28	1800
08	4.5	22	1200	31	6.5	16	600
09	4.5	22	1500	32	6.5	16	800
10	4.5	22	1800	33	6.5	16	1200
11	4.5	28	600	34	6.5	16	1500
12	4.5	28	800	35	6.5	16	1800
13	4.5	28	1200	36	6.5	22	600
14	4.5	28	1500	37	6.5	22	800
15	4.5	28	1800	38	6.5	22	1200
16	5.5	16	600	39	6.5	22	1500
17	5.5	16	800	40	6.5	22	1800
18	5.5	16	1200	41	6.5	28	600
19	5.5	16	1500	42	6.5	28	800
20	5.5	16	1800	43	6.5	28	1200
21	5.5	22	600	44	6.5	28	1500
22	5.5	22	800	45	6.5	28	1800
23	5.5	22	1200				

Initially, the first sample was positioned relative to a screen representing the luminous intensity diagram illustrated in Fig. 5. The photometer was placed at the first angle, and the measurement of the luminous intensity was registered. This procedure was repeated until the luminous intensity value referring to last angle of the sample was registered. Figure 6 illustrates this procedure for the pair of angles (0^oH, 5^oU) shown in Fig. 5.

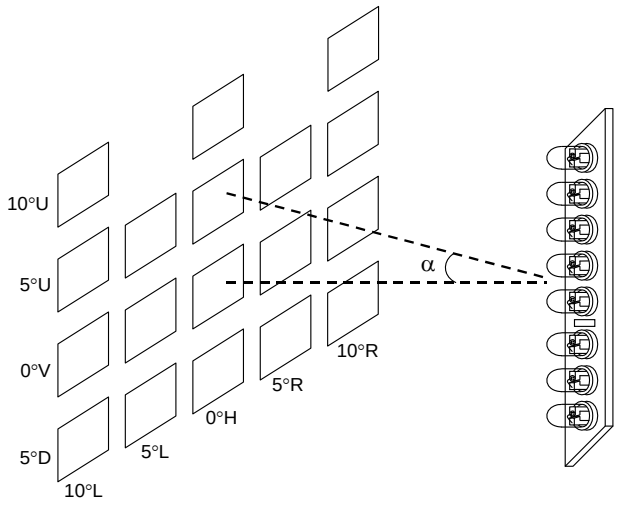


Fig 6. Luminous intensity diagram in relation to the brake light prototype.

The sample was then removed from the device, and a new sample was attached in order to measure the luminous intensity; all procedures are repeated until the value of the last angle of the last sample was registered.

From the design data provided in Table 1 and the measurement results of luminous intensity of brake light samples from different angles, a multilayer perceptron network was trained as will described. During this stage, a variation of the main network parameters was achieved. The number of layers, number of neurons per layer, activation function for each layer, and type of training were changed in order to obtain a neural network topology that could generate an acceptable mean squared error and ensure an efficient generalization. The best neural architecture for the simulations was selected by means of a cross-validation technique [10].

The topology chosen consisted of two hidden layers with 5 neurons in the first layer and 10 neurons in the second layer. The training algorithm was Levenberg-Marquardt [9]. The main advantage of this algorithm arises from its ability to accelerate the neural network convergence process, and it is considered to be the fastest method for training moderate-sized perceptron networks. In comparative terms, the Levenberg-Marquardt algorithm is about 100 times faster than the backpropagation method. For our application, the network inputs were defined by the 3 main parameters involved in brake light design, i.e.

- Distance between SMD LEDs → d (mm).
- Number of SMD LEDs → n .
- Luminous intensity of SMD LED → I_V (mcd).

The network output is composed by a unique signal which provides what is the intensity level produced by the brake light in a particular angle, i.e.

- Luminous intensity of brake light → I_{VBL} (cd).

After training, using the 18 different angles, one training for each angle, it was possible to estimate the total luminous intensity produced by the brake light in different angles. To validate the proposed approach are used data coming from samples not used in the network training. A comparison between the estimated values by the network and those provided by experimental tests is accomplished to analyze the efficiency of the proposed approach.

5 Results and discussion

The computational implementations of the neural networks used in this application were carried out using the software Matlab/Simulink. After the training process, the neural modeling was used to obtain luminous intensity values of brake lights, as previously described. Figure 7 illustrates a comparison between luminous intensity values (I_{VBL}) obtained by experimental tests (ET) and those estimated by the artificial neural network (ANN). In this configuration (Situation I), the used sample presents distance (d) between SMD LEDs equal to 5.5 mm, the number of SMD LEDs (n) is equal to 28 and the luminous intensity of each SMD LED (I_v) has a value equal to 800 mcd.

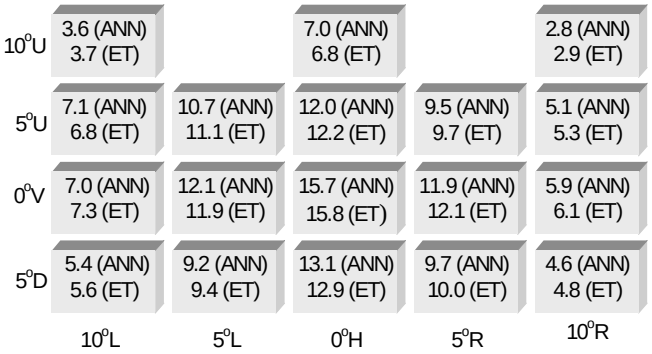


Fig. 7. Comparative illustration (Situation I).

From Fig. 7 it is observed that the generalization produced by the network to estimate values of luminous intensity in several angles is very satisfactory. In this case, the mean relative errors calculated were around 2.8% and with variance of 1.19%.

Figure 8 illustrates another comparison between luminous intensity values (I_{VBL}) obtained by experimental tests (ET) and those estimated by the artificial neural network (ANN). For this configuration (Situation II), the used sample presented the same distance (d) and the same SMD LEDs number of the previous situation; but, the luminous intensity of each SMD LED (I_v) has a value equal to 1200 mcd.

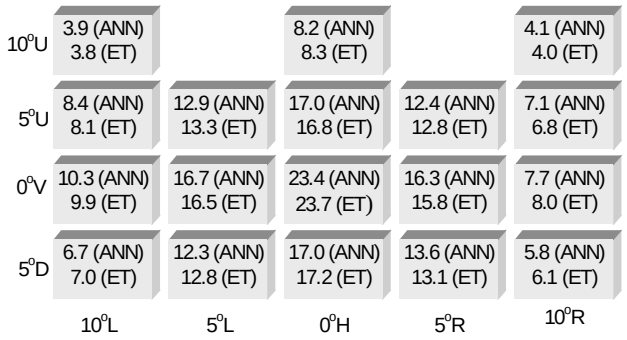


Fig. 8. Comparative illustration (Situation II).

In this case (Situation II), the mean relative errors calculated were around 3.0% and with variance of 1.61%.

Figure 9 illustrates another comparison between luminous intensity values (I_{VBL}) obtained by experimental tests (ET) and those estimated by the artificial neural network (ANN). In this configuration (Situation III), the used sample presents distance (d) between SMD LEDs equal to 6.5 mm, the number of SMD LEDs (n) is equal to 22 and the luminous intensity of each SMD LED (I_v) has a value equal to 1200 mcd.

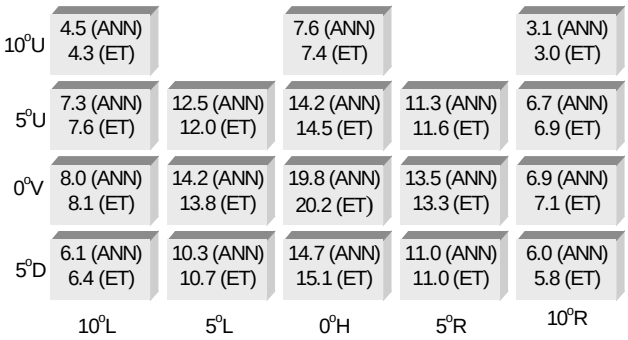


Fig. 9. Comparative illustration (Situation III).

In this case (Situation III), the mean relative errors calculated were around 2.9% and with variance of 1.47%.

Through these results it is possible to infer that the network presented efficient results for estimation of luminous intensity values of brake lights. It should be taken into account that the proposed neural network has considered the main parameters involved with the design of brake lights. In the selection process of the best neural architecture used in simulations was adopted the cross-validation technique [10].

6 Conclusions

This work presents a technique based on use of artificial neural networks for determination of luminous intensity values for brake lights, in which are considered the main design characteristics. Therefore, the developed tool constitutes a new technique that can efficiently be applied in this type of problem.

The developed methodology can also be generalized and used in other applications that use groups of SMD LEDs, such as in headlights, turning lights, rear lights, traffic lights, electronic panels of messages, etc.

The developed tool has significantly contributed for reduction of costs in relation to implementation stage of brake lights, i.e, it minimizes spent time in prototype designs. The tool has also allowed simulating many options for configurations of brake lights without need of building them, which also assists in the selection process of sample that offers an appropriate relationship between cost and benefit.

References

[1] S. B. Peralta, H. E. Ruda. "Applications for advanced solid-state lamps". IEEE Industry Applications Magazine, vol. no. 4, pp. 31-42, 1998.

[2] P. R. Edwards, R. W. Martin, I. M. Watson, C. Liu, R. A. Taylor, J. H. Rice, J. W. Robinson, J. D. Smith. "Quantum dot emission from site-controlled InGaN/GaN micropyramid arrays". Applied Physics Letters, vol. 85, pp 4281-4283, 2004.

[3] J. Voelcher. "Top 10 tech cars". IEEE Spectrum, vol. 41, pp. 20-27, 2004.

[4] W. R. Young, W. Wilson. "Efficient electric vehicle lighting using LEDs". Proc. of Southcon, pp. 276-280, 1996.

[5] B. G. Streetman, S. Banerjee. Solid State Electronic Devices. Prentice Hall, Englewood Cliffs, New Jersey, 1999.

[6] R. W. Martin, P. R. Edwards, R. A. Taylor, J. H. Rice, J. W. Robinson, J. D. Smith, C. Liu, I. M. Watson. "Luminescence properties of isolated InGaN/GaN quantum dots". Physica Status Solidi (A), vol. 202, pp. 372-376, 2005.

[7] R. Pecharroman-Gallego, R. W. Martin, I. M. Watson. "Investigation of the unusual temperature dependence of InGaN/GaN quantum well photoluminescence over a range of emission energies". Journal of Physics D: Applied Physics, vol. 21, pp. 2954-2961, 2004.

[8] P. Griffiths, D. Langer, J. A. Misener, M. Siegel, C. Thorpe. "Sensor-friendly vehicle and roadway systems". 18th Instrumentation and Measurement Technology Conference, pp. 1036-1040, 2001.

[9] M. T. Hagan, M. B. Menhaj. "Training feedforward networks with the Marquardt algorithm". IEEE Transactions on Neural Networks, vol. 6, pp. 989-993, 1994.

[10] S. Haykin. Neural Networks. Prentice-Hall, Upper Saddle River, New Jersey, 1999.

[11] S. Kim, S. -Y. Oh, J. Kang, Y. Ryu, K. Kim, S. -C. Park, K. Park. "Front and rear vehicle detection and tracking in the day and night times using vision and sonar sensor fusion". IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2173-2178, 2005.

[12] I. Cabani, G. Toulminet, A. Bensrhair. "Color-based detection of vehicle lights". IEEE Intelligent Vehicles Symposium, pp. 278-283, 2005.

[13] P. Mottier. LED for Lighting Applications. Wiley-ISTE, Hoboken, New Jersey, 2009.

Voltage Sags/Swells Mitigation Using a Dynamic Voltage Restorer Controlled by Neural Network

Julio César Suárez-Duarte, Elisa Espinosa-Juárez
Electrical Engineering Faculty, Universidad Michoacana de San Nicolás de Hidalgo
Morelia, Michoacán, México

Abstract - *In this paper the effectiveness of a Dynamic Voltage Restorer controlled by neural network for voltage sags/swells mitigation is shown. A typical configuration of DVR is implemented by using MATLAB Simulink®, which is controlled using backpropagation neural networks. The good performance for voltage sags/swells mitigation of the DVR controlled with neural networks can be observed by means of several examples. Simulation results are also compared with the results obtained by using another DVR control technique.*

Keywords: Dynamic Voltage Restorer, neural networks, voltage sags/swells, power quality

1 Introduction

Power quality is referred as the characteristics and suitable conditions of the electrical energy supplied to equipments and devices that allow to maintain the electrical continuity, without affecting the performance and also without causing faults in the components.

The poor quality in the energy supply affects, in mayor or in minor grade, several devices and consequently the industrial process, where the economical losses related with this aspect could be very important [1][2].

Voltage sags are one of the most important problems in the electrical supply, which are defined as a decrease of *rms* voltage, between 0.1 and 0.9 p.u., with duration of 1/2 cycle to 1 minute [3]. Voltage sags origin is associate with faults in the electrical network (mainly caused by lightening, accidents, or operation mistakes); voltage sags are also caused by overloads, starting of large motors, connection/disconnection of large loads. The effects on voltage at buses are the decrease of voltage magnitude and also the phase angle jump, and this conditions could affect the regular operation of equipment, even could cause damage in equipments, causing the interruption of manufacture process, leading to time and production losses, and consequently, causing significant economical losses [2].

The most severe voltage sags are the ones caused by faults in the electrical network, their residual magnitude depends on the network and the nearness of the fault to the bus where the voltage sag is observed [1]. Voltage sags are mainly related with random events, consequently, their occurrence in the electrical system could change greatly over a period of time.

The temporary overvoltages or voltage swells are defined as the increase of the *rms* voltage, between 1.1 and 1.8 p.u. with duration of 1/2 cycle to 1 minute [3]. Such disturbances are not as frequent as voltage sags; voltage swells occurring with the disconnection of large capacitor banks or large loads, but usually are associate with system unbalanced faults, for example, during a single phase to ground fault a swell can occur on the unfaulted phases [4].

The most common method to the mitigation of voltage sags/swells is the installation of additional equipment on the network-equipment interface. Recent developments are addresses to this form of mitigation. The popularity of the mitigation equipment is explained because with the installation of this type of devices the customer has the whole control of the situation. The changes in the energy supply as the improved of equipment in the network are usually out of control of the final user. This method of mitigation includes several devices based on power electronic, the Voltage Source Converter (VSC) which is a dispositive that can generate a voltage of a desired magnitude, frequency and phase angle. In mitigation of voltage sags/swells is used to replace temporarily the voltage supply or to provide voltage in order to compensate the lost voltage during the disturb [1].

The same technology of the VSC is used in the known Flexible Alternating Current Transmission Systems (FACTS) [5] [6]. FACTS devices, such as Static Compensator (STATCOM) and Dynamic Voltage Restorer

(DVR) are widely used for mitigation of voltage sags/swells and other disturbances [1].

The proper functioning of the previously mentioned devices depends largely on the control system that is used. Recently, the use of artificial intelligence techniques in power electronics and the control field has increased considerably. This methods, based on experts systems, the fuzzy logic and neural networks, generally need a large data base or knowledge that describe the operation of the system with a logical analysis, instead of a mathematic analysis to control the system [7].

In this paper the application of artificial neural networks in the control system of a DVR is presented; its proper performance for mitigation of voltage sags/swells at load buses of an electrical network is shown.

The paper is organized as follows. In Section 2 the DVR typical arrange, its topology and control is briefly explained. In Section 3 the DVR control system by applying neural networks is described, and the implementation in MATLAB Simulink® is explained. In Section 4 study cases considering voltage sags and swells disturbances are presented, and the effectiveness of the DVR with the system control using neural networks is shown.

2 Typical topology for Dynamic Voltage Restorer

The basic operation of a DVR is to control the voltage applied to the load injecting a voltage of appropriate characteristic, in order to maintain the nominal voltage in the load even when a disturbance is present in the supply network.

The DVR consist of a VSC connected in series with the power supply by means of a coupling transformer (see Fig. 1), the voltage in the connections of the load is the sum of the voltage of the grid and the voltage injected by the DVR, this is

$$v_L = v_S + v_{DVR} \quad (1)$$

The DVR is constituted of the following elements [8]:

1) Voltage Sourced Converter which consist on:
   Stored energy unit. It is used to supply energy when the compensation of voltage is required. The DC stored energy can be delivered from different kinds of energy storage systems, such as batteries, capacitors, among others.

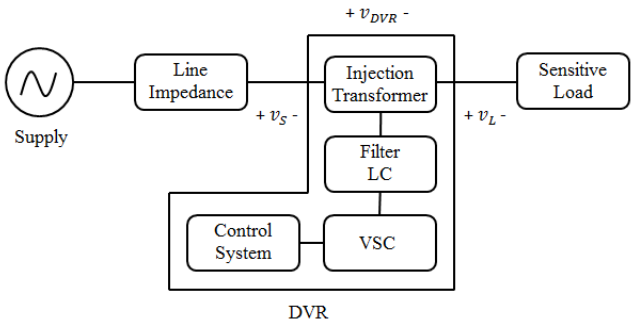


Fig. 1 Tipycal DVR configuration

   Three-phase voltage inverter. Generally, the DVR uses Insulated Gate Bipolar Transistors (IGBT) in a structure of voltage three-phase inverter controlled by techniques of Pulse-Width Modulation (PWM). In this work, the technique of Sinusoidal Pulse Width Modulation (SPWM) is used to control the three-phase voltage inverter.
2) Passive LC filter. It is a circuit of inductances L, and capacitances C, used to remove the high frequencies in the inverter output.
3) Injection three-phase transformer. It is a three-phase transformer which with its low voltage winding connected in series with the line, while the high voltage winding is connected to the LC filter output coming from the three-phase inverter. It is important to notice that the injection transformer could cause voltage drops that must be considered in the voltage control of the DVR.

The amount of stored energy depends on the power given by the inverter and the maximum duration of the voltage sag/swell. Typically, the controller is designed to operate in certain range within a maximum to a minimum duration of the event. Some practical aspects about DVR are discussed in [9].

It is important to mention that there are others topologies of DVR in which there is not stored energy unit and the injected power is provided through a transformer from the grid (source side or load side) via a rectifier [8][10].

3 Implementation of the DVR control by applying neural networks

Artificial neural networks were initially used for images and sounds recognition, data and signal processing, and to classification. Nowadays, the neural networks are applied in diverse knowledge fields, such as power electronic and control system. In the control systems, the expectation of

improve several operation aspects has driven research about the application on neural networks in this field.

This type of structures are organized in a groups of neurons that process the information of the inputs in parallel and then the outputs of the neurons can be combined to obtain network outputs or to feed other neurons group.

In this work the DVR and its control system based on neural networks has been implemented in MATLAB Simulink®, using the Neural Network Toolbox® [11].

A backpropagation neural network is made up of several layers of neurons connected in cascade, having the ability to auto adjust the weights of neurons in the intermediate class to learn the relationship between a dataset inputs/outputs, which are known as training set.

Characteristics and the process of setting up the backpropagation neural network used in this work can be summarized as follows:

1) Implementation of a typical configuration of the DVR, in order to obtain operation dataset.
2) Creation of training dataset with voltage measurements of voltage disturbances (inputs) and the voltage in the sensitive load (outputs) for different operating conditions. To this end, voltage measurements were taken both at the source phase as at the load.
   From the measurements of voltage on the source, the transformation of three-phase voltage into *dq0* voltage is obtained in order to accurately detect the beginning of the disturbance [12]. In the load side, peak voltages are measured for each phase, and the difference between the desired nominal voltage at the load is used to form the dataset. This form of neural network training has the advantage that the voltage drop across the injection transformer and other parameters such as the LC filter at the converter output will not affect the desired voltage applied to the load.
   Several fault condition were simulated by means of a three-phase programmable voltage source, generating a input/output training dataset considering variations of ramp type, from 0.1 p.u. to 1.8 p.u. in order to cover values in the sags and swells threshold.
3) Neural network creation. The feed-forward backpropagation neural network with three layers is created by using *newff* function.
4) Neural network training. From the obtained dataset multiple input/output sets are selected to train the neural network by using *trainlm* function.
5) Testing neural network. After carrying out the neural network training the behavior of the neural network is tested with different inputs to ensure that the training has been satisfactory.

4 Simulation of DVR

Next, in order to illustrate the operation of the control scheme based on neural networks for the DVR, described in previous section, several studies are carried out using the implemented arrangement in the MATLAB Simulink® environment.

Fig. 2 shows the implementation of the electrical grid connected to the load where the DVR is included. The circuit has a programmable voltage source, which can simulate different fault conditions. The RL sensitive load is connected to the supply through the DVR.

Fig. 3 shows the typical configuration of a DVR composed of a DC source as energy storage unit, a three-phase inverter composed of IGBTs, an LC filter, bypass switches and a three-phase injection transformer.

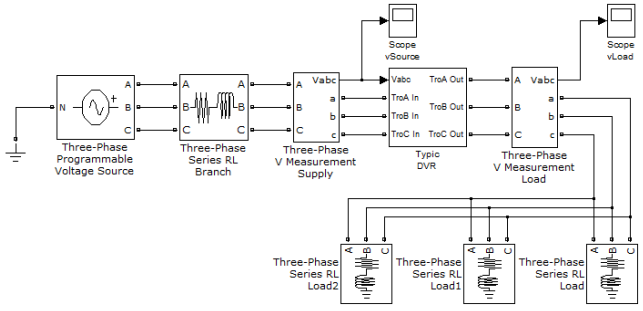


Fig. 2 Electrical grid supplying energy to the load trough the DVR

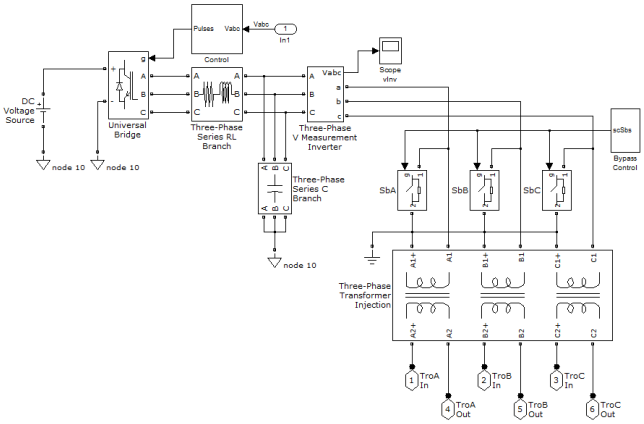


Fig. 3 Typical DVR with its coupling three-phase transformer which inject the voltage in series with the supply

Fig. 4 shows the implementation of the control system based on neural networks for the DVR. As before mentioned, the neural network have as input *dq0* values and as output voltage peak values $(\hat{V}_a, \hat{V}_b, \hat{V}_c)$ so that an additional block is included in the output of these networks to provide the three-phase voltage of reference to the PWM. This is,

$$\begin{aligned} v_a &= \hat{V}_a \sin \omega t \\ v_b &= \hat{V}_b \sin \omega t + 240° \\ v_c &= \hat{V}_c \sin \omega t + 120° \end{aligned} \qquad (2)$$

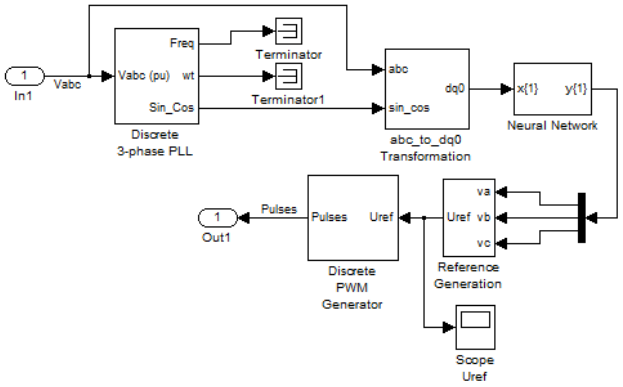


Fig. 4 Implemented DVR control based on neural networks

The simulation parameters considered in the implementation are shown in the Appendix.

Fig. 5 shows the performance of the DVR when a voltage sag with residual magnitude of 0.5 p.u. occurs in the power supply. In Fig. 5a) the voltage supply is shown. The voltage sag starts at 0.1s and stay up 0.2s. Fig. 5b) and Fig. 5c) show the voltage injected by the DVR controlled by neural network and the corresponding load voltage with compensation. As a result of DVR action, the load voltage is the desired nominal voltage (1 p.u.).

Fig. 6 shows the performance of the DVR when a swell occurs in the power supply. In Fig. 6a) the swell of 50% over the nominal voltage is shown. Fig. 6b) and 6c) show the injected and the load voltage respectively. Similar to the case of voltage sag, the load voltage remains in 1 p.u.

As can be seen from the results, the voltage in the load stay up in the nominal value because the DVR operates almost instantly to inject the appropriate voltage component in order to correct the supply voltage.

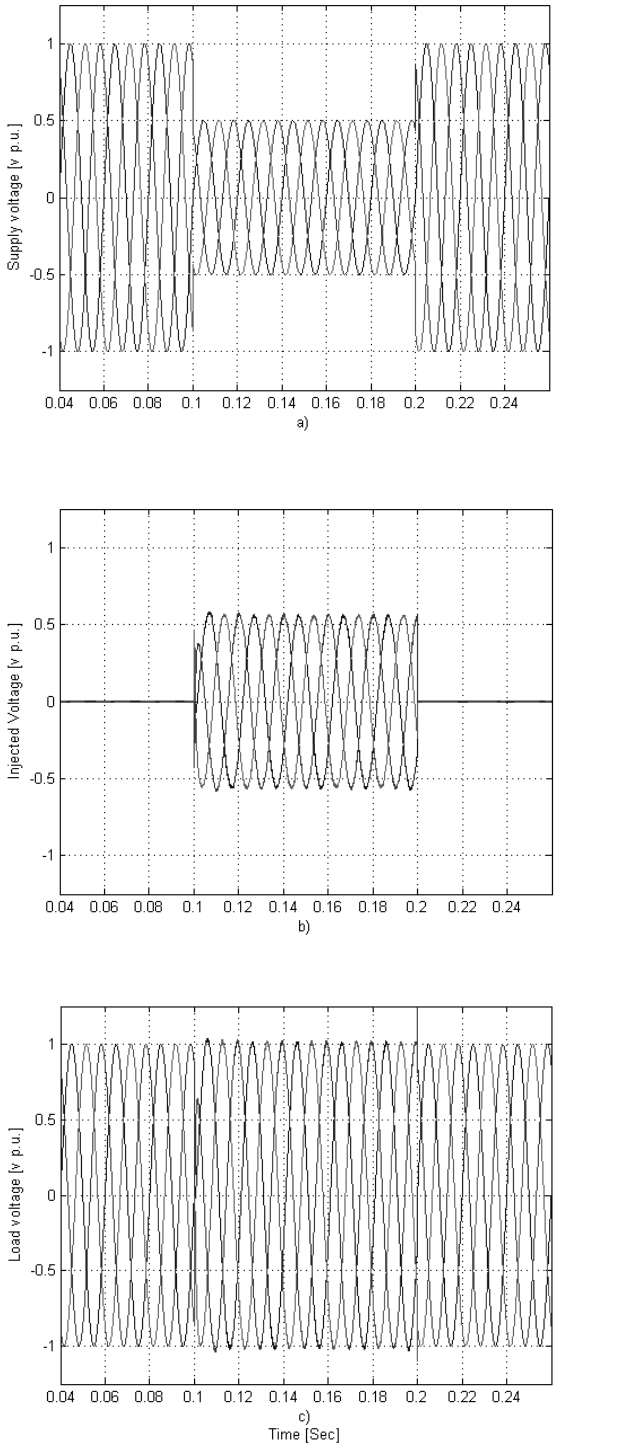


Fig. 5 Three-phase voltage sag with residual magnitude of 0.5 p.u.: a) Voltage supply; b) Injected voltage by the DVR controlled by neural network; c) Load voltage.

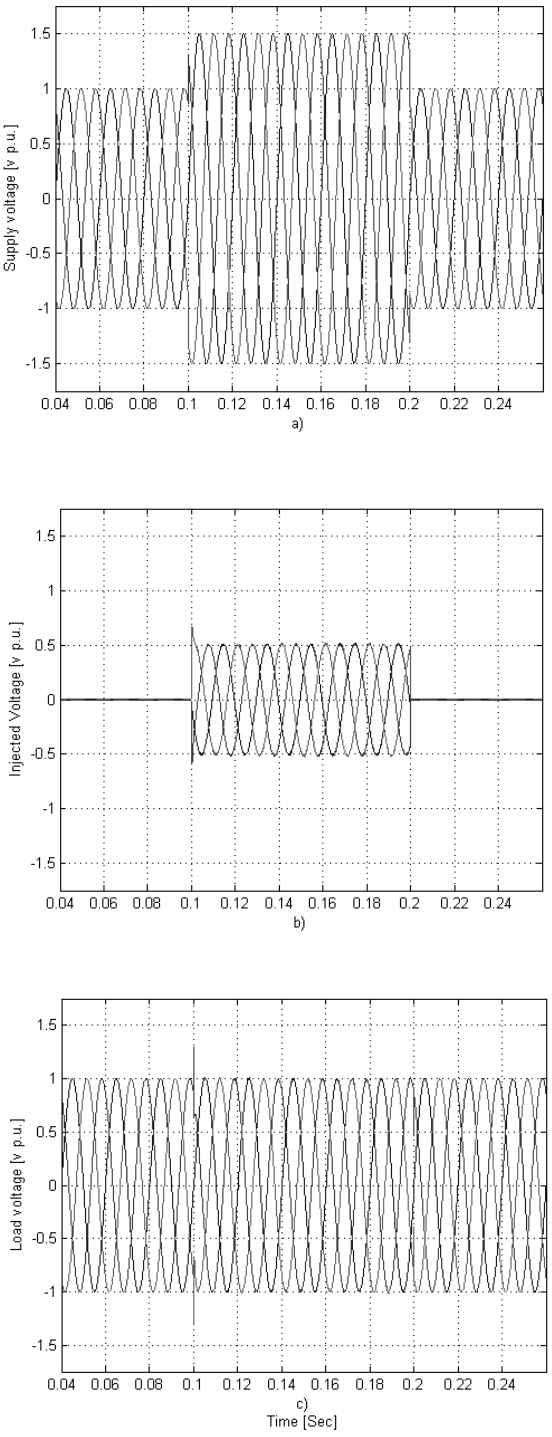


Fig. 6 Three-phase voltage swell with magnitude of 1.5 p.u.: a) Voltage supply; b) Injected voltage by the DVR controlled by neural network; c) Load voltage.

In Fig. 7-9 the performance of DVR controlled with neural network scheme is compared with respect to the performance of a DVR controlled with *dq0* transformation scheme proposed in [13] which has been also implemented in Matlab Simulink®. The DVR controlled with *dq0* transformation works well for certain magnitude range of sags and swells, this does not occurs for any disturbance magnitude.

Fig. 7 shows the voltage supply when a voltage sag with residual magnitude of 0.8 p.u. occurs. In Fig. 8 can be observed that the DVR controlled with neural network works correctly when the 0.8 p.u. voltage sag occurs, however if the DVR is controlled with *dq0* transformation (implemented according to [13]) the injected voltage is not enough for the complete mitigation of the voltage sag (see Fig. 9). This can be because in the simulations ideal elements are not considered which cause differences between the real voltage in the load and the desired nominal voltage.

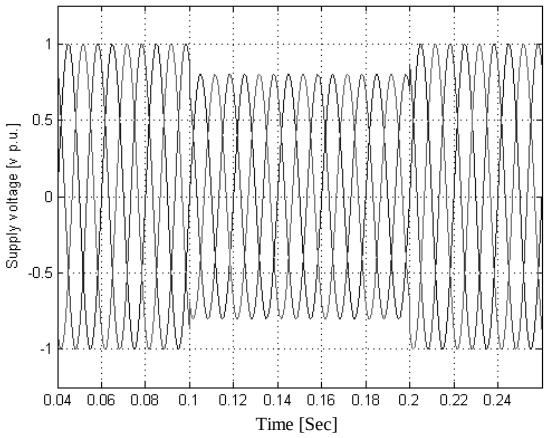


Fig. 7. Voltage supply: three-phase voltage sag with residual magnitude of 0.8 p.u.

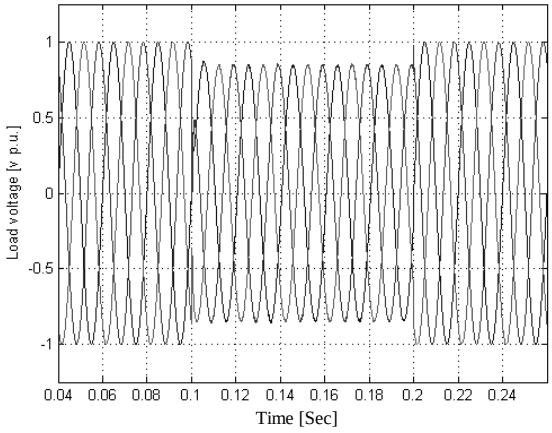


Fig. 8. Load voltage when a voltage sag of 0.8 p.u. residual magnitude occurs, and the DVR is controlled with *dq0* method.

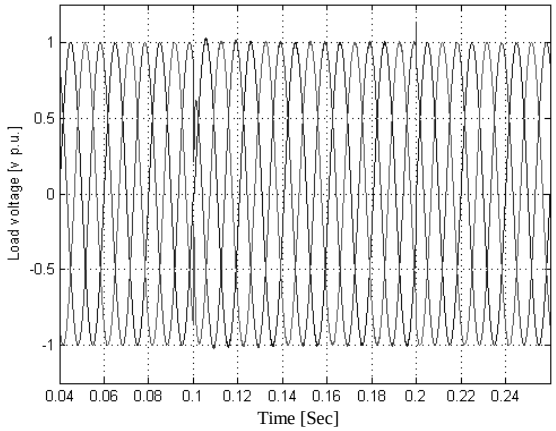


Fig. 9. Load voltage when a voltage sag of 0.8 p.u. residual magnitude occurs, and the DVR is controlled by neural networks.

5 Conclusions

In this paper an electrical system connected to a sensitive load through a DVR controlled with neural network has been implemented. The effectiveness of the proposed and implemented control system in the mitigation of voltage disturbances such as sags and swells has been shown.

The performance of the proposed control based on neural networks is compared control technique based on *dq0* transformation of the three-phase voltage.

From the simulations carried out it can be observed that when a realistic characterization of the elements in the implemented electrical system is considered, the DVR controlled with neural network has a better performance.

6 Appendix

SIMULATION PARAMETERS DATA

Voltage (line to line)	200 V
Frequency	50 Hz
DC voltage	800 V
Line resistance	1e-3 Ω
Line inductance	1e-6 H
Filter resistance	0.1 Ω
Filter inductance	8e-3 H
Filter capacitance	11e-6 F
Transformer injection turns ratio	1:1
Sensitive Load	4250 W
Power factor	0.85
Sample time	5e-6 s

7 Acknowledgment

This work has been partially supported by the National Council of Science and Technology of México (CONACYT) under Grant 335179/233249.

8 References

[1] M. H. J. Bollen. "Understanding Power Quality Problems". IEEE Press Series on Power Engineering, 2000.

[2] J. Cardona, J. Barros. "ATP Prediction of Voltage Sag Effect on Sensitive Loads"; ICREPQ International Conference on Renewable Energy and Power Quality, April 2006.

[3] IEEE Standard 1159-1995. "IEEE Recommended Practice for Monitoring Electric Power Quality", 1995.

[4] R. C. Dugan, M. F. McGranagham, S. Santoso, H. W. Beaty. "Electrical Power Systems Quality". McGrawHill, 2004.

[5] C. Schauder, M. Gerhardt, E. Stacey, T. W. Cease, A. Edris, T. Lemak, L. Guygyi. "Development of a 100 MVar Static Condenser for Voltage Control of Transmission Systems"; IEEE Transactions on Power Delivery, Vol. 10, No. 3, pp. 1486-1496, July 1995.

[6] L. Guygyi, C. Schauder, S. L. Williams, T. R. Rietman, D. R. Torgerson, A. Edris. "The Unified Power Flow Controller: A New Approach to Power Transmission Control"; IEEE Transactions on Power Delivery, Vol. 10, No. 2, pp. 1085-1097, April 1995.

[7] V. G. Kinhal, P. Agarwal, H. O. Gupta. "Performance Investigation of Neural-Network-Based Unified Power-Quality Conditioner"; IEEE Transactions on Power Delivery, Vol. 26, No. 1, pp. 431-437, January 2011.

[8] J. G. Nielsen, F. Blaabjerg. "A Detailed Comparison of System Topologies for Dynamic Voltage Restorers"; IEEE Transactions on Industry Applications, Vol. 41, No. 5, pp. 1271-1280, September/October 2005.

[9] S. W. Middlekauff, E. R. Collins. "System and Customer Impact: Considerations for Series Custom Power Devices"; IEEE Transactions on Power Delivery, Vol. 13, No. 1, pp. 278-282, January 1998.

[10] E. Babaei, F. K. Mohammad, S. Mehran. "Mitigation of Voltage Disturbances Using Dynamic Voltage Restorer Based on Direct Converters"; IEEE Transactions on Power Delivery, Vol. 25, No. 4, pp. 2676-2683, October 2010.

[11] Neural Network Toolbox 2: "User's Guide", The MathWorks, Inc., 2008.

[12] Y. H. Chung, H. J. Kim, G. H. Kwon, T. B. Park, S. H. Kim, K. S. Kim, J. I. Moon. "Medium Voltage Dynamic Voltage Restorer with Neural Network Controlled Voltage Disturbance Detector"; International Conference on Power System Technology, PowerCon 2006, October 2006.

[13] R. Omar, N. A. Rahim, M. Sulaiman. "Modeling and Simulation for Voltage Sags/Swells Mitigation Using DVR"; Journal of Theoretical and Applied Information Technology, pp. 464-470, 2009.

Biographies

Julio César Suárez-Duarte received the B. E. degree in Electrical Engineering from the Universidad Michoacana de San Nicolás de Hidalgo (UMSNH), México. He is currently working toward the M.Sc. Degree in electrical engineering at the UMSNH. His research interests include power quality and power electronics.

Elisa Espinosa-Juárez received the B. E. degree in electrical engineering from the Universidad Michoacana de San Nicolás de Hidalgo (UMSNH), México, in 1986, the M.Sc. degree in electrical engineering from the Instituto Politécnico Nacional, México D.F., México, in 2001, and the Ph. D. degree in electrical engineering from the Universidad Politécnica de Madrid (UPM), Madrid, Spain, in 2006.

Currently, she is an Associate Professor with the UMSNH, Morelia, México. Her research interests include power quality and, especially, voltage sags.

Cascade-Correlation Neural Networks for Breast Cancer Diagnosis

A. Nachev[1], **M. Hogan**[1], **and B. Stoyanov**[2]
[1]Business Information Systems, Cairnes Business School, NUI, Galway, Ireland
[2]Dept. of Computer Science, Shumen University, Shumen, Bulgaria

Abstract - *This study explores the predictive abilities of the cascade-correlation neural networks as a tool for breast cancer diagnosis. The dataset used for training and testing contains a combination of mammographic, sonographic, and other descriptors, which is novel for the field. We applied feature selection techniques to find an optimal set of descriptors that ensure high sensitivity and specificity. The model performance was estimated by ROC analysis and metrics derived from it, such as max accuracy, full and partial area under the ROC curve and the convex hull, and specificity at 98% sensitivity. Our findings show that particular feature selection techniques applied with the cascade-correlation model outperform the traditional backpropagation networks in all the metrics. The proposed model also provides advantages, such as self-organization of the structure, few parameters to adjust, and fast training, which makes it a better alternative for applications in the domain.*

Keywords: cascade correlation, neural networks, breast cancer diagnosis, data mining, CAD.

1 Introduction

Earlier treatment of breast cancer requires early diagnosis, and early diagnosis requires an accurate and reliable diagnostic procedure that allows physicians to differentiate benign from malignant lesions. Some studies show that only a third of suspicious masses are determined to be malignant and many surgical biopsies are unnecessary due to high false positive rates of indication for the disease (Jemal et al., 2005), (Lacey et al., 2002). The low efficiency of the diagnosis unnecessary exposes patients to discomfort, anxiety, complications, and can distort future mammograms. In addition, the financial burden of these procedures (thousands of euros each) is significant in the present political and economic effort to reduce expenditures.

Computer-aided diagnosis (CAD) of the disease is a typical classification problem which was approached over the years by many techniques and methods based on variety of sources of medical information, such as digitized screen-film mammograms, full-field digital mammograms, sonograms, magnetic resonance imaging (MRI) images, and gene expression profiles (Jesneck et al., 2006), (Zonderland et al., 1999), (Vyborny, 1994). Currently, CAD implementations tend to use only one information source, usually mammographic data in the form of descriptors of the Breast Imaging Reporting and Data System (BI-RADS). BI-RADS is a lexicon developed in 1993 by the American College of Radiology (ACR) in order to standardize the mammographic language and interpretations, and to facilitate communication between clinicians (Kopans, 1992). Initially, BI-RADS had been applied to mammography only and did not pertain to other breast imaging techniques, but recently ACR developed a BI-RADS lexicon for breast sonography, which standardizes characterization of sonographic lesions such as mass shape, orientation, margin, and posterior acoustic transmission, and other sonographic features (ACR, 2003), (Jackson, 1995).

Jesneck et al. (2007) have used a specific combination of BI-RADS mammographic and sonographic descriptors and some proposed by Stavros et al. (1995) to build a predictive model. Their study was pioneering in using such a combination and they report that predictive abilities of the linear discriminant analysis (LDA) and multi-layer preceptrons (MPL) (Rumelhart, 1986) are similar in the context of using either all 39 descriptors or a suggested subset of 14 descriptors. MLP have been largely applied to breast cancer diagnosis applications, but they have a drawback: the model assumes predefined network architecture, including connectivity and node activation functions, and training algorithm to learn to predict. The issue of designing a near optimal network architecture can be formulated as a search problem and still remains open.

Many CAD applications have shown themselves promising for improving sensitivity (true positive rate, or hit rate), but few of them achieve improved specificity (false positive rate, or false alarm rate). The purpose of this study is to explore the predictive power of the cascade-correlation networks used with the combination of descriptors discussed above and estimate the model characteristics in terms of both sensitivity and specificity.

The paper is organized as follows: Section 2 provides an overview of the cascade-correlation neural network architecture used build a predictive model; Section 3 discusses the dataset used in the study, its features, preprocessing steps, and feature selection; Section 4 presents and discuses the experimental results; and Section 5 gives the conclusions.

2 Cascade-Correlation Neural Networks

Cascade-correlation neural networks (Fahlman and Libiere, 1990) are supervised self-organizing networks with structure similar to backpropagation networks. Instead of adjusting the weights in a network of fixed topology, a cascade-correlation net begins with a minimal number of nodes, then automatically trains and adds new hidden nodes one by one and do not change them over the time. It creates a multi-layer structure called a 'cascade' because the output from all input and hidden nodes already in the network feed into new nodes.

A cascade-correlation net has three layers: input, hidden and output (Figure 1). Initially, the network begins with only input and output neurons. The *input layer* is used to accept the variable values ($x_1 \ldots x_p$) presented to it and without processing distributes them to the neurons in the hidden and output layers. In addition to the input variables, there is a bias input of signal 1 and weight θ, similarly to the bias input of the backpropagation networks.

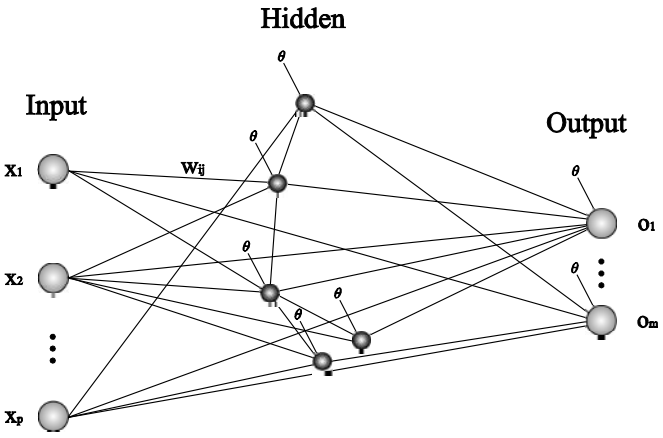


Figure 1. *A cascade-correlation neural network with input, hidden and output layers.*

The *output layer* consists of a single neuron if the network is used for regression problems, or contains several neurons for classification problems, one per class label. Each output neuron receives values from all of the input neurons, including the bias, and all of the hidden layer neurons. Each value presented to an output neuron is multiplied by a weight, and the resulting weighted values are added together producing a combined value. The weighted sum is fed into a neuron activation function, which outputs a value – the network output. For classification problems, a linear activation function is used in the output neurons; regression problems require a sigmoid activation function.

The *hidden layer* is empty in the beginning – every input is connected to every output neuron by a connection with an adjustable weight. Such a simple cascade-correlation network has considerable predictive power and for a number of applications it provides excellent predictions. If not, however, the network adds new hidden nodes one by one (see Figure 2) until the residual error gets acceptably small or the user interrupts this process.

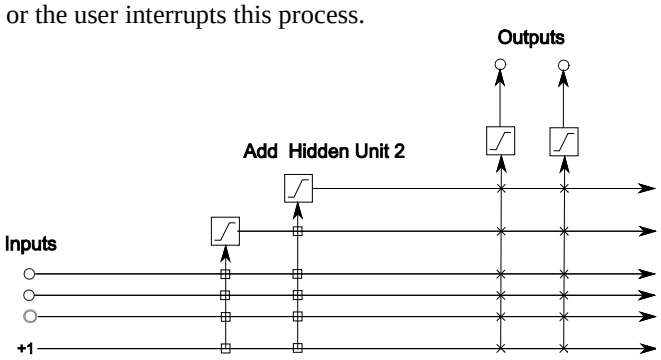


Figure 2. *Cascade architecture after adding two hidden nodes (adapted from (Fahlman & Lebier, 1991)). The vertical lines sum all incoming activation. Boxed connections are frozen, 'x' connections are trained repeatedly.*

To install a new hidden neuron, instead of a single candidate, the system uses a pool of trainable candidate nodes (usually four to eight), each with a different set of randomly selected initial weights. All candidates receive the same input signals and see the same residual error for each training pattern, but because they are not installed yet and do not interact with one another or affect the active neural network during training, all of these candidate units are trained in parallel; when no further progress is being made in training, the network installs the candidate whose score is the best (minimises the residual error). The use of this pool of candidates is beneficial in two ways: it greatly reduces the chance that a useless unit will be permanently installed, and it speeds up the training because many parts of weight-space can be explored simultaneously.

While the candidate weights are being trained, none of the weights in the active network are changed. Once a new hidden node has been added to the network, its input-side weights (boxed connections in Figure 2) are frozen; the output-side connections ('x' connections) continue to be adjustable. The learning algorithm modifies the weights attempting to minimize the residual error of the network. Each new neuron becomes a permanent feature-detector in the network, available for producing outputs or for creating other, more complex feature detectors.

The new hidden neuron receives inputs from the external inputs and pre-existing hidden neurons. The values on a vertical line in Figure 2 are added together after being multiplied by their weights so that each output neuron receives a weighted sum from all of the input neurons including the bias θ_i (1).

$$s_i = \sum_j w_{ij} x_j + \theta_i \quad , \tag{1}$$

The hidden and candidate nodes are usually of the same type, with a sigmoid activation function (2)

$$O_i(s_i) = \frac{1}{1+e^{-\beta s_i}} \quad , \tag{2}$$

where β is slope parameter. Alternatively, a mixture of nodes with different nonlinear activation functions can be used, such as sigmoid, Gaussian, radial and so on. The resulting networks, with a mixture of functions adapt specifically to a particular application problem and may lead to more compact and elegant solutions than are possible in homogeneous networks.

The cascade-correlation architecture has several advantages over the traditional backpropagation neural nets. First, as the network is self-organizing and determines its own size and topology during the growth of the hidden layer during training, there is no need to decide how many layers and neurons to use in the network. This is a major problem of the backpropagation networks which implies use of predefined network architecture and designing a near optimal network architecture is a search problem that still remains open. Secondly, cascade-correlation nets learn very quickly (often 100 times as fast as a backpropagation network) and retain the structures they have built even if the training set changes. This makes them suitable for applications with large training datasets. Cascade-correlation nets are typically small, with fewer than a dozen neurons in the hidden layer in contrast to probabilistic neural networks which require a hidden-layer neuron for each training pattern. Finally, the training of the cascade-correlation networks is quite robust, and good results usually can be obtained with little or no adjustment of parameters. They have less chance to get trapped in local minima compared to the backpropagation nets.

As with all types of models, the cascade-correlation networks feature some disadvantages. They have a significant potential for overfitting the training data which results in a very good accuracy on the training dataset but poor accuracy on new, unseen during the training data. Cascade-correlation networks are usually outperformed by probabilistic and general regression neural networks on small to medium size problems with up to a couple of thousand training patterns, but they are superior in handling large problems far better than probabilistic or general regression networks.

3 Dataset and Preprocessing

We use a dataset collected from 2000 to 2005 at Duke University Medical Centre (Jesneck et al., 2007) which contains mammographic and sonographic findings from physical examination of patients, information about patient family history of breast cancer, and personal history of breast malignancy. The data set contains 803 patterns of which 296 malignant and 507 benign. Each record contains 39 attributes – 13 are mammographic BI-RADS, 13 are sonographic BI-RADS, 6 are sonographic suggested by Stavros et al. (1995), 4 are sonographic mass descriptors, and 3 are patient history.

The features are as follows: mass size, parenchyma density, mass margin, mass shape, mass density, calcification number of particles, calcification distribution, calcification description, architectural distortion, associated findings, special cases (as defined by the BI-RADS lexicon: asymmetric tubular structure, intramammary lymph node, global asymmetry, and focal asymmetry), comparison with findings at prior examination, and change in mass size. The sonographic features are radial diameter, antiradial diameter, anteroposterior diameter, background tissue echo texture, mass shape, mass orientation, mass margin, lesion boundary, echo pattern, posterior acoustic features, calcifications within mass, special cases (as defined by the BI-RADS lexicon: clustered microcysts, complicated cysts, mass in or on skin, foreign body, intramammary lymph node, and axillary lymph node), and vascularity. The six features suggested by Stavros (Stavros et al., 1995) are mass shape, mass margin, acoustic transmission, thin echo pseudocapsule, mass echogenicity, and calcifications. The four other sonographic mass descriptors are edge shadow, cystic component, and two mammographic BI-RADS descriptors applied to sonography—mass shape (oval and lobulated are separate descriptors) and mass margin (replaces sonographic descriptor angular with obscured). The three patient history features were family history, patient age, and indication for sonography.

Neural network applications usually require preprocessing of the train and test data where variable values differ significantly because they are different in nature or they have been measured by different units. For example, the mass size and the calcification values are between 0-3 and 0-75 respectively. Such an inconsistency can lead to poor classification results as some variables can dominate over others. The preprocessing addresses this problem by normalization and rescaling transformations. The purpose of the normalization (3) is to make each variable (separately) have zero mean and unit standard deviation. The rescaling (5) scales down the each variable within the unit hypercube.

$$\tilde{x}_i = \frac{x_i - \overline{x}_i}{\sigma_i}, \tag{3}$$

where

$$\overline{x}_i = \frac{1}{N}\sum_{n=1}^{N} x_i^n \ , \qquad \sigma_i^2 = \frac{1}{N-1}\sum_{n=1}^{N}(x_i^n - \overline{x}_i)^2 \tag{4}$$

$$x_i^{new} = \frac{x_i^{old} - \min_i}{\max_i - \min_i} \tag{5}$$

where min_i and max_i are the max and min values of the variable x_i.

We separated the dataset into two parts by using 5-fold cross-validation: 80% of the patterns were used for training and validation purposes using; 20% were used to test the

model performance given that they were never presented to the network for training.

Using all features of a dataset does not always lead to best or even satisfactory results because too much information presented to the network can lead to overfitting / overtraining. There are feature selection techniques that can produce subsets of features with high abilities to discriminate between the classes. This also helps to build a robust learning model, speed up learning process, improve the model interpretability, and help to understand which features are important and what are relationships between them.

Theoretically, best feature selection technique is the exhaustive search among all possible subsets of features, but this is impractical for datasets with high cardinality (39 in our case). A better alternative approach is to use subset selection algorithms or feature ranking techniques, but the former are preferable as usually provide better results. We applied genetic search, best first search, subset size forward selection, race search, and scatter search. Our results show that the subset size forward selection, proposed by Guetlien et al. (2009) gives best results with our model. The algorithm is an extension of the linear forward selection that reduces the number of attributes expansions at each forward selection step. The technique is faster, finds smaller subsets and increases accuracy compared to the standard forward selection. The subset it outputs (s17) contains 17 descriptors, namely: patient age, indication for sonography, mass margin, calcification number of particles, architectural distortion, anteroposterior diameter, mass shape, mass orientation, lesion boundary, special cases, mass shape, mass margin, thin echo pseudocapsule, mass echogenicity, edge shadow, cystic component, and mass margin. Two of these are general descriptors; three - mammographic BI-RADS; five - sonographic BI-RADS; four - Stavros'; and three - sonographic mass descriptors. The feature set is relatively balanced in representing different categories of data. We also considered a feature set of 14 descriptors (s14) proposed by Jesneck et al. (2007) and obtained by stepwise feature selection. In our experiments we also included the full set of 39 features (s39).

4 Experiments and Discussion

Using the data described above we built and tested a predictive model based on a cascade-correlation neural network with Turboprop2 learning algorithm, which is a variant of the Fahlman's cascade-correlation learning (Fahlman and Libiere, 1990). Turboprop2 dynamically grows hidden neurons and trains the network very fast preventing the model from overfitting. In order to avoid bias in training and testing we did 5-fold cross-validation as discussed above, which means that five copies of the classifier were tested on 20% of the dataset after training it on the remainder. Results were averaged and grouped in four categories: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). For model estimation we calculated accuracy Acc=(TP+TN)/(TP+TN+FP+FN); true positive rate TPR=TP/(TP+FN), and false positive rate FPR=FP/(TP+FN).

Each of the feature sets s14, s17, and s39 was tested to see how it discriminates between benign and malignant cases and the results show that best performer is s17 (84.8% accuracy), followed by s14 (83.8%) and s39 (82.8%). Compared with a backpropagation MLP trained with the same dataset in the same experimental context (Nachev & Stoyanov, 2010), the cascade-correlation model shows improvement in accuracy nearly 3%. This however does not provide the full picture of the model and its features. In machine learning accuracy is the most common and valuable estimator, but it can be insufficient or misleading when the application requires specific operational parameters and the misclassifications have different cost. This is the case of the application domain discussed here, because typically radiologist and practitioners operate with high sensitivity levels (above 90%) and the cost of misclassified malignant and benign lesions is different with different consequences (Jiang et al., 1996).

The receiver operating characteristics (ROC) analysis provides an alternative approach for estimation (Fawcet, 2006). It considers relation between two indices: TPR (sensitivity) and FPR (1- specificity) as shown in Figure 3. The cascade-correlation model is a soft classifiers as it outputs values between 0 and 1, which are to be mapped to true / false by using a threshold that determines the operating point at which the clinical decision is made. Varying the threshold results in many classifiers (points) on the ROC space. The curve is a step function that depicts the trade-off between sensitivity and specificity for every possible threshold. In terms of the application domain, the threshold value represents the operating point above which a lesion is considered suspicious for malignancy and recommended for biopsy, otherwise biopsy is not recommended. The best possible prediction model would be the most north-west point of the curve as it is closest to the ideal classifier with coordinates (0, 1) having 100% sensitivity and 100% specificity. From another hand a random guess would give a point on the no-discrimination line. Figure 3 shows three ROC curves of the cascade-correlation net trained and tested with the three data sets s14, s17, and s39.

The ROC curve is also a base for construction of the ROC convex hull (ROCCH) – a convex broken line that connects the most northwest points along the ROC curve (compared to their neighbors) as well as the two trivial classifiers (0, 0) and (1, 1).

As the ROCCH contains all possible optimal classifiers and an optimal classifier must lay on it, all classifiers below the ROCCH can be ignored. Additionally, a section of the ROCCH line represents a continuum of possible classifiers not produced by experiments, but could be constructed by a linear combination (randomly weighting) of the two 'real' classifiers represented by the end points of that section.

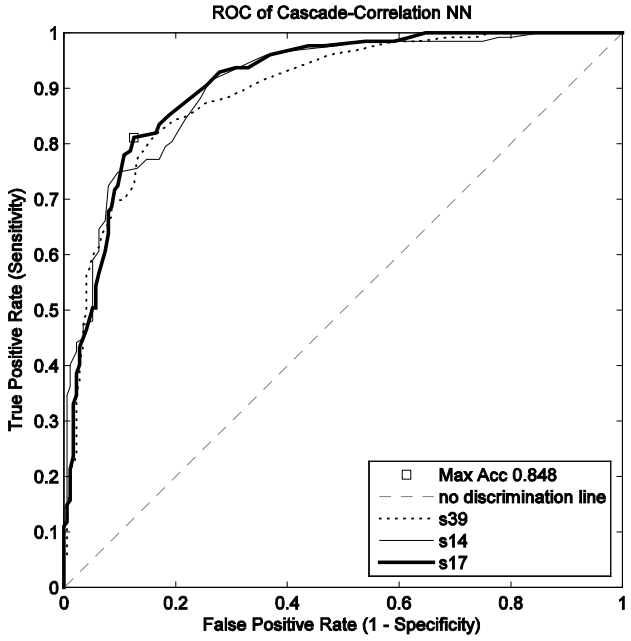


Figure 3. *ROC space of cascade-correlation neural net applied to three feature sets: s14, s17, and s39.*

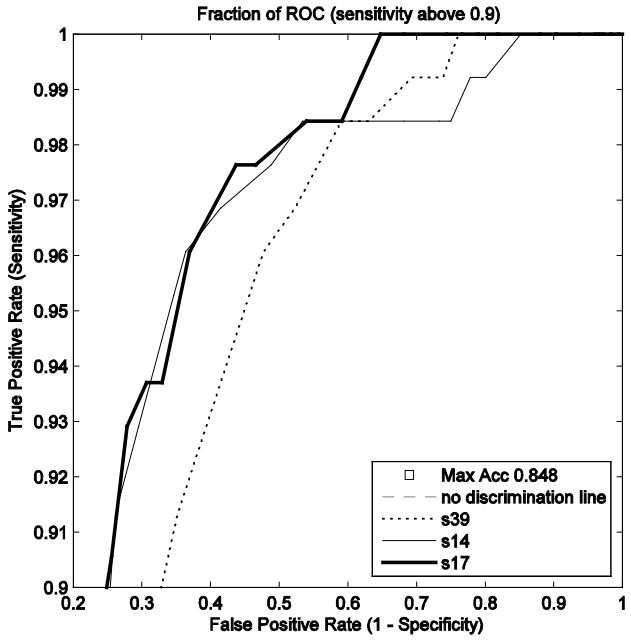


Figure 4. *Fraction of the ROC space of cascade-correlation network with high sensitivity (above 90%). Curves represent performance of the three feature sets: s14, s17, and s39.*

The ROC analysis also allows to use metrics that can quantify a model performance, such as area under the ROC curve (AUC); area under the convex hull (AUH); partial AUC / AUH ($_{p}$AUC / $_{p}$AUH) which take into account only the area where sensitivity is between p and 1 (p=0.9 as clinically relevant); and specificity at 98% (nearly perfect) sensitivity. Figure 4 shows a fraction of the ROC space where sensitivity is between 0.9 and 1.

The metrics can be interpreted as follows: AUC represents the overall model performance in general regardless the decision threshold; $_{0.90}$AUC shows the overall model performance relevant to the application as it requires high sensitivity; specificity at 98% sensitivity is another important metric from a clinical viewpoint. Table 1 shows the experimental results comparing the three feature sets.

According to the figures best overall performer is the cascade-correlation model trained and tested with s17 (AUC=0.911), followed by s14 and s39. The AUC exceeds that of the backpropagation MLP (Nachev & Stoyanov, 2010), but the difference is not significant. In terms of the clinically relevant metrics, however, the model shows some significant differences. The $_{0.90}$AUC exceeds that of the MLP 5% (73% vs. 68%) and the model outperform the radiologist practices by 21% (73% vs. 52%) (Jesneck et al., 2007). The other metric - specificity at 98% sensitivity shows superiority of the cascade-correlation over the traditional backpropagation in all three datasets with improvement 5%, 1%, and 22% respectively and nearly reaching the radiologist performance.

Table 1 Performance metrics of cascade-correlation neural network with three feature sets.

Performance Metric	*39 attr. Set*	*17 attr. set*	*14 attr set.*	*Radiologist assessment*
AUC	0.896	0.911	0.907	0.92
$_{0.90}$AUC	0.648	0.68	0.731	0.52
Spec at 98% sens	0.427	0.5	0.49	0.52
AUH	0.901	0.916	0.916	n/a
Acc_{max}	0.828	0.848	0.838	n/a

Finally, it can be summarized that the cascade-correlation neural networks appear to be a better alternative for building a predictive model for diagnosis in the application domain compared to the traditional backpropagation networks. It also provides promising clinically relevant characteristics comparable with the radiologist activities and practices.

5 Conclusions

Only one third of all breast cancer biopsies made today confirm the disease, which makes the economic and social cost of this kind of diagnosis very high. This study explores the predictive abilities of the cascade-correlation neural networks as a core of computer-aided diagnosis tools. The dataset contains 803 patterns of 39 BI-RADS mammographic and sonographic descriptors and some proposed by Stavros et al. (1995). Using such a combination of features is novel for the field. We applied feature selection

techniques to reduce the data dimensionality and to find an optimal set of descriptors that can ensure high sensitivity and specificity of the model. We quantified the model performance by using ROC analysis and metrics derived from it, such as max accuracy, full and partial area under the ROC curve and the convex hull, and specificity at 98% sensitivity. Our findings show that the cascade-correlation model outperforms the traditional backpropagation networks in all of those metrics. The model also features both high sensitivity and specificity which would allow decreasing unnecessary benign breast biopsies while minimizing the number of delayed breast cancer diagnoses. From another hand, the cascade-correlation networks have several advantages over the classic backpropagation neural nets, such as self-organization of the structure, few parameters to adjust, fast and robust training with little chance to get trapped into local minima. These characteristics render the cascade-correlation nets as a promising technique for applications in the field.

6 References

[1] American College of Radiology. BI-RADS: ultrasound, 1st ed. In: Breast imaging reporting and data system: BI-RADS atlas, 4th ed. Reston, VA: American College of Radiology, 2003

[2] Fahlman, S. and Lebiere C. "The Cascade-Correlation Learning Architecture" in D. S. Touretzky (ed.), Advances in Neural Information Processing Systems 2, Morgan Kaufmann, 1990.

[3] Fawcett, T. "An introduction to ROC analysis"; Pattern Recognition Letters, Vol. 27 Issue 8, pp. 861-874, 2006.

[4] Guetlein, M., Frank, E., Hall, M., Karwath, A. "Large Scale Attribute Selection Using Wrappers"; In Proc. IEEE Symposium on CIDM, pp.332-339, 2009.

[5] Horsch, K., Giger, M., Vyborny, C., Venta, L. "Performance of computer-aided diagnosis in the interpretation of lesions on breast sonography." AR, vol. 11, pp. 272–280, 2004.

[6] Jackson, V., "Management of Solid Breast Nodules: What is the Role of Sonography?"; Radiology, Vol. 196, pp. 14–15, 1995.

[7] Jesneck, J., Nolte, L., Baker, J., Floyd, C., Lo, J. "Optimized Approach to Decision Fusion of Heterogeneous Data for Breast Cancer Diagnosis."; Medical Physics, Vol. 33, Issue 8, pp.2945-2954, 2006

[8] Jesneck, J., Lo, J., Baker, J. "Breast Mass Lesions: Computer-Aided Diagnosis Models with Mamographic and Sonographic Descriptors"; Rad., 244:2, pp 390-398, 2007.

[9] Jiang, Y., Metz, C., Nishikawa, R. "A ROC partial area index for highly sensitive diagnostic tests."; Radiology, vol. 201, pp.745-750, 1996.

[10] Kopans D. "Standardized mammographic reporting"; Radiol Clin North Am, Vol. 30, pp. 257–261, 1992

[11] Lacey, J., Devesa, S., Brinton, L. "Recent Trends in Breast Cancer Incidence and Mortality."; Environmental and Molecular Mutagenesis, Vol. 39, pp. 82–88, 2002.

[12] Nachev, A. and Stoyanov, B.: An Approach to Computer Aided Diagnosis by Multi-Layer Preceptrons. In Proc. of International Conference Artificial Intelligence, Las Vegas (2010)

[13] Rumelhart, D., McClelland, J. "Parallel Distributed Processing"; Explorations in the Microstructure of Cognition, Cambridge, MA: MIT Press, 1986.

[14] Stavros, A., Thickman, D., Rapp, C., Dennis, M., Parker, S., Sisney, G. "Solid Breast Modules: Use of Sonography to Destinguish between Benign and Malignant Lesions"; Radiology, Vol. 196, pp. 123-134, 1995.

[15] Vyborny, C. "Can Computers Help Radiologists Read Mammograms?" Radiology, Vol. 191, pp. 315-317, 1994.

[16] Zonderland, H., Coerkamp, E., Hermans, J., Van de Vijver, M., Van Voorthuisen, A. "Diagnosis of Breast Cancer: Contribution of US as an Adjunct to Mammography."; Radiology, Vol. 213, pp. 413–422, 1999.

PREDICTION OF GROUNDWATER LEVELS USING DIFFERENT ARTIFICIAL NEURAL NETWORK ARCHITECTURES AND ALGORITHMS

P. Sujatha[1] and Dr. G.N. Pradeep Kumar[2]
1. Associate Professor in Civil Engg. Dept. of Mechanical Engg., JNTUH College of Engg., Jagtyal-505501, Karimnagar district, India
2. Professor, Department of Civil Engg, S.V. University, Tirupati-517 507 Andhra Pradesh, India

ABSTRACT

Performance of four types of functionally different artificial neural network (ANN) models, namely Feed forward neural network, Elman type recurrent neural network, Input delay neural network and Radial basis function network and fourteen types of algorithms, namely Batch gradient descent (traingd), Batch gradient descent with momentum (traingdm), Adaptive learning rate (traingda), Adaptive learning rate with momentum (traingdx), Resilient backpropagation (trainrp), Fletcher-Reeves update (traincgf), Polak-Ribiere update (traincgp), Powell-Beale restarts (traincgb), Scaled conjugate gradient (trainscg), BFGS algorithm (trainbfg), One step secant algorithm (trainoss), Levenberg-Marquardt (trainlm), Automated regularization (trainbr) and Random order incremental training (trainr) were examined in order to identify an efficient ANN architecture and algorithm that can simulate the water table fluctuations using a relatively short length of groundwater level records. Tirupati, located in Chittoor district of the drought-prone Rayalaseema region in India, having resident population of over 3.0 lakhs and pilgrims of over 50,000 per day was chosen as the study area. As its groundwater levels showed a rapid decline in the last decade due to the overexploitation for the domestic, agricultural and industrial needs, accurate prediction is very essential to plan better conservation of groundwater resources. Results showed that Feed forward neural network trained with training algorithm Levenberg-Marquardt is suitable for accurate prediction of groundwater levels.

Keywords: Neural Network, Groundwater level forecasting, Evapotranspiration

INTRODUCTION

Artificial Neural Network (ANN) technique is recently attracting considerable attention in all branches of engineering and science. ANN offers the distinctive ability to learn complex nonlinear relationships without requiring the mechanistic knowledge about the underlying systems. Therefore, it has a great potential in areas such as hydrological systems where complex, dynamic and highly nonlinear mechanisms are the norm. ANNs were used for rainfall-runoff modeling, precipitation forecasting, groundwater modeling, water quality modeling, water management, reservoir operations and other hydrologic applications (ASCE, 2000a, b; Coulibaly et al, 2001a, b, c; Ioannis N.Daliakopoulos et al, 2004; Purna C.Nayak et al, 2005). In this study, an attempt was made to compare the performance of four types of functionally different ANN models and fourteen types of algorithms that can best predict the groundwater levels.

Groundwater is one of the most valuable natural resources for many developing countries, as it is reliable in dry seasons or droughts, cheaper to extract, requires little treatment and is less affected by catastrophic events. In the more arid areas, where rainfall is low, groundwater may be the only source of supply for domestic, agricultural and industrial purposes. Without pro-active management and protection, there is a serious risk of deterioration of groundwater levels on an increasingly widespread basis. Under the pressure of the need to rapidly develop new water supplies, there is rarely adequate attention to, and investment in, the maintenance, protection and long-term sustainability of groundwater. In this context, a reliable water supply planning policy necessitates accurate prediction of groundwater levels. This generally requires a longer data of water table depth measurements, which are usually unavailable. Therefore a common approach is to use empirical time series models as described by Box and Jenkins (1976) and Hipel and McLeod (1994) to generate a longer period of water table depths. Unfortunately, the major disadvantage of empirical models is its inaccuracy in prediction when the dynamical behavior of the hydrological system changes with respect to time (Bierkens, 1998). In general, relationships between the precipitation, the nearby surface water and the groundwater are nonlinear. However, owing to the

difficulties of identifying nonlinear model structure and estimating the associated parameters, only very few nonlinear empirical models, such as stochastic differential equation and threshold autoregressive self-exciting open-loop models, have been proposed for shallow water table modeling (Bierkens, 1998; Knotters and De Gooijer, 1999). Alternatively, one can resort to physical models (Belmans et al., 1983; Feddes et al., 1988). But, in practice, requirement of data to simulate water table fluctuation are enormous and tedious, hence not affordable for developing countries. Therefore a dynamical predictive model is very desirable for improved water resources management and reliable water supply planning where groundwater remains the primary source of water.

METHODOLOGY

In its most general form, an ANN is a technique that is designated to model the way in which the brain performs a particular task. To achieve good performance, ANNs employ a massive interconnection of neurons. The architecture of a network describes how many layers a network has, the number of neurons in each layer, each layer's transfer function and how the layers connect to each other. Feed forward neural network, Elman type recurrent neural network, Input delay neural network and Radial basis function network are some of the widely used ANN architectures. Figure 1a shows feed forward network. The main advantage of feed forward neural networks is that they are easy to handle, and can approximate any input/output map (Hornik et al. 1989). Elman network is a two-layer network with feedback from the first layer output to the first layer input and allows both to detect and generate time-varying patterns. Figure 1b shows Elman type recurrent neural network. Input delay neural network consists of a memory structure and a nonlinear associator (Figure 1c). Radial basis function networks are non linear hybrid networks containing a single hidden layer of computation nodes. Figure 1d shows the network consisting of a hidden layer of R nodes, r inputs and one output. Net input to the hidden layer is the vector distance between its weight vector w and the input vector p, multiplied by the bias b.

An algorithm is a procedure for solving a problem, in which a list of well-defined instructions for completing a task will proceed through a well-defined series of successive states, eventually terminating in an end-state. Traingd, Traingdm, Traingda, Traingdx, Trainrp, Traincgf, Traincgp, Traincgb, Trainscg, Trainbgf, Trainoss, Trainlm, Trainbr and Trainr are the often used algorithms in the model development. In order for an ANN to generate an output vector $Y=(y_1, y_2,\ldots\ldots y_p)$ that is as close as possible to the target vector $T=(t_1, t_2,\ldots\ldots t_p)$, a training process is employed to find optimal weight matrices W and bias vectors V that minimize a predetermined error function that has the form:

$$E = \sum_{P} \sum_{p} (y_i - t_i)^2 \quad \ldots \quad (1)$$

Here, t_i is a component of the desired output T; y_i is the corresponding ANN output; p is the number of output nodes and P is the number of training patterns. The primary goal of training is to minimize the error function by searching for a set of connection strengths and threshold values that cause the ANN to produce outputs that are equal or close to targets. The process of optimizing the connection strengths and threshold values can be carried out using local or global methods. Local methods fall into two categories: first-order and second-order methods. First-order methods are based on a linear model (gradient descent) whereas second order models are based on a quadratic model (e.g., Newton's method) (Battiti, 1992). In both cases, iterative techniques are used to minimize the error function. Description of various algorithms which deal with adjustment of connection strengths and threshold values can be obtained from any standard books on neural networks.

In the present study, an attempt was made to compare the performance of four types of functionally different ANN models and fourteen types of algorithms that can best simulate the groundwater levels.

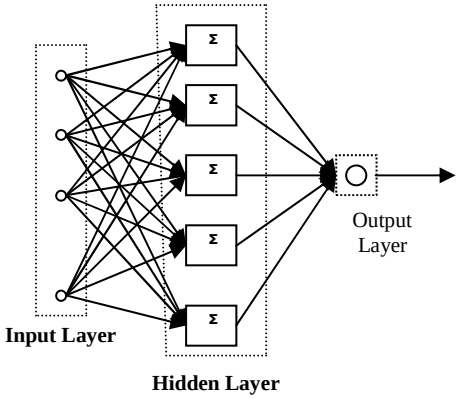


Figure 1a. Feed forward Neural Network

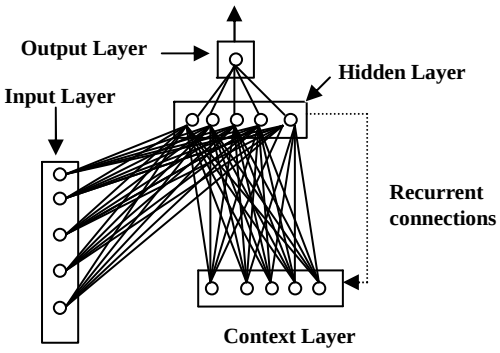


Figure1b. Elman type Recurrent Neural Network

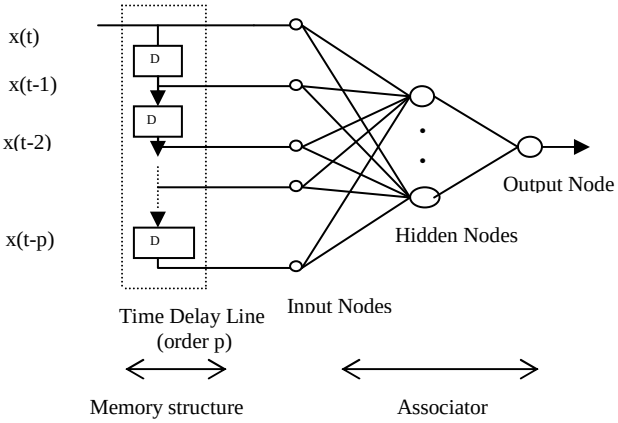


Figure 1c. Input delay neural network with p memory order

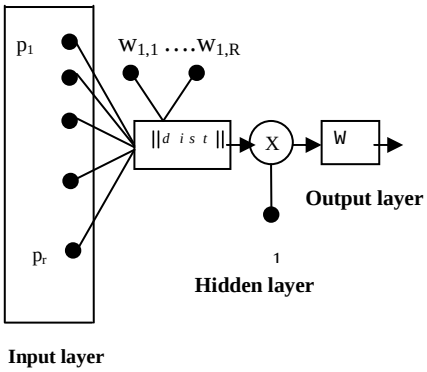


Figure1d. Radial Basis function network

STUDY AREA and DATA

All the four architectures and algorithms are tested with data taken from observation wells located in Tirupati. Figure 2 shows the Area of Study. Tirupati is world renowned pilgrim centre, connected to various parts of the globe through railways, roadways and airways. It has a static population of more than 3.0 lakhs in an area of 110 km^2, in addition to a daily pilgrimage of 50,000 to 60,000. This figure touches to 1.0 lakh during summer vacation and Brahmotshavams. The climate of Tirupati, on the whole, is dry. The annual mean rainfall is 700mm. Geologically, Tirupati region is covered by granites and dyke rocks of Archean age overlain by recent alluvium deposits of 10-25m thick. The pace of urban development and the rapid increase in population of Tirupati town, which started in the 1980's, has led to the depletion of surface and groundwater resources. The large numbers of visitors with their additional demands for water has further aggravated the problem of water scarcity. Available surface water resource from the Kalyani Dam and Telugu ganga is not adequate to cope up with the needs. Further, due to successive failure of monsoons, water supply from bore wells has become an important resource. Heavy dependence on groundwater has resulted in increasing numbers of wells and boreholes. Hence, prediction of groundwater levels reliably with the available data of shorter length necessitates the administrators to plan for the demands of the future. In this study, four different types of neural network architectures and algorithms were examined to determine the network that is best suitable to predict the water table fluctuations for the observation wells located in Tirupati.

Figure 2. Location of study area

MODEL DEVELOPMENT

Input Variable Selection: In order to develop a suitable neural network for predicting the groundwater levels, appropriate assignment of past values of inputs to input nodes is required. Basically, the crucial process of developing a predictive model is to identify the input variables among the available variables for each output variable. Generally, sensitivity analysis is carried out to assess the relative significance of each input variable. A glance at the past work (Coulibaly et al, 2001a; Ioannis N.Daliakopoulos et al, 2004; Purna C.Nayak et al, 2005) indicates that past precipitation data; minimum, maximum and mean temperatures; stream flow and groundwater level data were considered as inputs. In the present study, precipitation and past groundwater levels for the period 1999 to 2006, along with evapotranspiration data were selected as input variables. Evapotranspiration data were also included as input variables to improve the model performance. FAO Penman-Monteith method was used for the estimation of evapotranspiration. Identified input variables for the observation wells located in Tirupati are R(t-1), R(t-2), GWL(t-1), GWL(t-2) & ET(t-1) where R is rainfall data, GWL is groundwater levels and ET is evapotranspiration, t-1 and t-2 are data pertaining to 1 and 2 lag time steps respectively. Data used for training the network is normalized to remove the cyclicity. Variables were scaled between -1 and 1. Number of hidden neurons in the proposed network was determined by several trails starting with two hidden neurons and increased upto 20 with a step size of one in each trial. Learning rate and momentum factors were so selected that they are responsive to the complexity of the local error surface. For each set of hidden neurons, network was trained with different architectures in batch mode to minimize the mean square error at the output layer. Training was stopped when there was no significant improvement in the efficiency, and then model was tested for its generalization properties. Optimum number of hidden neurons was found to be 5. A tansigmoidal function was used as the activation function in both hidden and output layers for Feed forward network, Elman type recurrent network, Input delay network and for all the fourteen algorithms. Groundwater level is treated as output variable. Therefore the proposed model has 5 input nodes with a single input layer, 5 hidden nodes with a single hidden layer and 1 output node in the output layer for all the networks studied. Using the error estimating functions, such as Correlation Coefficient (R^2), Average Absolute Relative Error (AARE) and Root Mean Square Error (RMSE), the effectiveness of each network is tested.

RESULTS AND DISCUSSION

Once the architecture of the model was optimized (i.e. 5 input nodes, 5 hidden nodes and 1 output node), the selected inputs R(t-1), R(t-2), GWL(t-1), GWL(t-2) and ET(t-1) were fed into the MATLAB 7.0 environment with Feed forward network, Elman type recurrent network, Input delay network and Radial basis network and for the fourteen algorithms for the proposed model. For training the model, data pertaining to the period between 1999 and 2004 (360 readings) were used. Model was tested using data pertaining to the period between 2005 and 2006 (120 readings). Efficiencies of the models were then tested using R^2, AARE and RMSE and values are listed in Table 1 for four architectures and in Table 2 for fourteen algorithms. The values of R^2 suggest that all the four models were able to train the network effectively. The lower values of AARE and RMSE also indicate that all the four models were able to forecast the groundwater levels efficiently during training. Values of R^2, AARE and RMSE suggest that Feed forward network showed the best performance in simulating the observed groundwater levels among the networks studied. The next best performance was observed by Input delay network followed by Elman type recurrent network. In contrast, Radial Basis function network was unable to predict the groundwater levels efficiently during the testing period. This suggests that the Radial basis networks may not be suitable for water table modeling based on short length calibration data. Radial basis networks are very sensitive to the presence of noise in the data and this could be the reason for this network to perform poorly in predicting the groundwater levels. These observations suggest that Feed forward network trained with training algorithm 'Levenberg-Marquardt' showed best performance in predicting the groundwater levels with data of relatively shorter period for the observation wells located in Tirupati.

Table 1: R^2, AARE and RMSE for the observation wells in Tirupati

Architecture	TRAINING			TESTING		
	R^2	AARE	RMSE	R^2	AARE	RMSE
Feed forward neural network	0.998	1.180469	0.24585	0.966	3.378442	1.271485
Elman type recurrent neural network	0.998	1.187832	0.259775	0.935	4.340135	1.342292
Input delay neural Network	0.998	1.162424	0.259749	0.947	3.924771	1.322815
Radial basis function Network	1.000	0.049615	0.02626	0.697	6.99177	2.336718

Table 2: R^2, AARE and RMSE for the observation wells in Tirupati

S.No.	Network Algorithm	TRAINING			TESTING		
		R^2	AARE	RMSE (m)	R^2	AARE	RMSE (m)
1	Traingd	0.998	1.238143	0.259781	0.956	4.095239	1.33899
2	Traingdm	0.998	1.218206	0.259797	0.951	4.274524	1.329271
3	Traingda	0.998	1.243446	0.259764	0.95	3.861408	1.306574
4	Traingdx	0.998	1.217718	0.259718	0.917	4.666426	1.403462
5	Trainrp	0.998	1.18792	0.449952	0.942	4.558326	1.440199
6	Traincgf	0.998	1.22249	0.259619	0.939	4.311618	1.335061
7	Traincgp	0.998	1.330689	0.259649	0.925	4.473834	1.375171
8	Traincgb	0.998	1.261698	0.259652	0.929	4.425914	1.344445
9	Trainscg	0.998	1.194846	0.259704	0.954	3.733864	1.273483
10	Trainbfg	0.998	1.224501	0.254332	0.944	3.831408	1.278989
11	Trainoss	0.998	1.229115	0.259508	0.923	4.527738	1.370561
12	Trainlm	0.998	1.180469	0.24585	0.966	3.378442	1.271485
13	Trainbr	0.999	0.826061	0.170469	0.931	4.083465	1.310148
14	Trainr	0.998	1.156513	0.260647	0.935	4.513809	1.372788

Observations from Table 2 suggest that Bayesian regularization algorithm (trainbr) and Levenberg-Marquardt algorithm showed better performance during training. Observations from table 2 during testing suggest that the best performance was achieved by feed forward network trained with 'Levenberg-Marquardt' (trainlm) algorithm. Though Bayesian regularization algorithm (trainbr) showed better performance during training, values of R^2, AARE and RMSE during testing indicate that it was not able to generalize with the same accuracy.

Figure 3 shows the forecasted groundwater levels and observed groundwater levels during testing period. It is clear that groundwater levels computed by the models Feed forward network, Elman type recurrent network and Input delay network are in good agreement with those of the observed groundwater levels. However deviation is noticed in the values of groundwater levels computed by Radial basis function network with the observed groundwater levels.

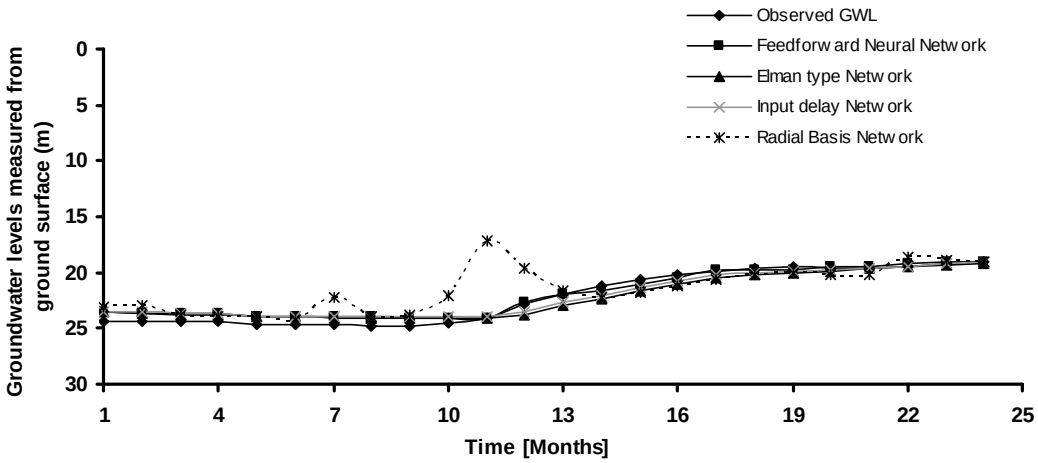


Fig. 3 Observed Groundwater levels and Groundwater levels computed by different network architectures during testing

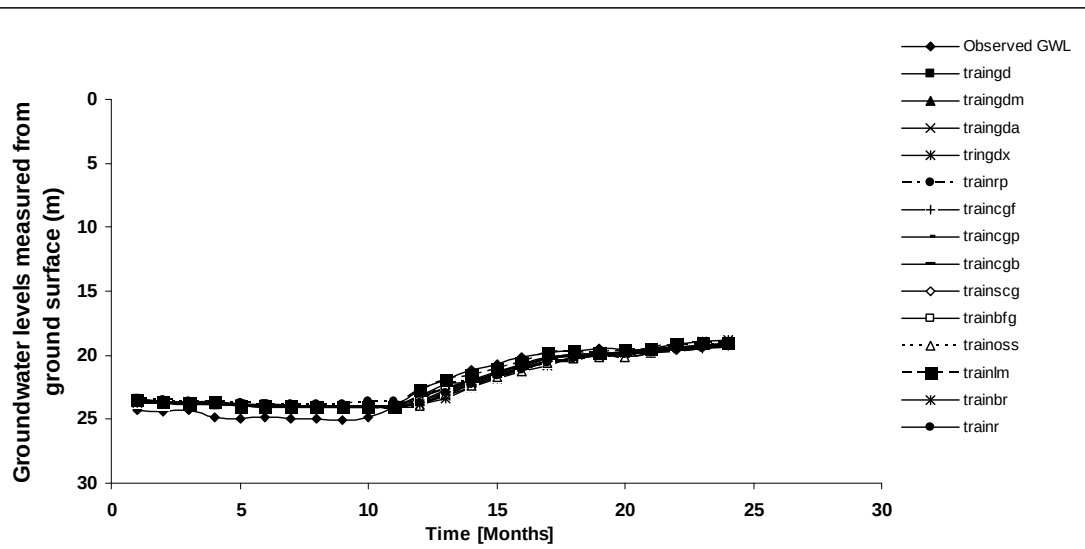


Figure 4. Observed Groundwater levels and Groundwater levels computed by different algorithms

Figure 4 shows the forecasted groundwater levels and observed groundwater levels during testing. It is clear that of all the 14 network algorithms, feed forward network trained with Levenberg-Marquardt algorithm was observed to predict the groundwater levels very efficiently and accurately.

CONCLUSIONS

In countries like India where vast area depends largely on groundwater resources, non-availability of groundwater level records for sufficient period has been the main hindrance in developing a model for prediction of realistic groundwater levels. In this paper an attempt was made for more realistic prediction of groundwater levels with the data of shorter period for the observation wells located in Tirupati, Chittoor district using ANN. Four different network architectures namely Feed forward network, Elman type recurrent network, Input delay network and Radial basis function network were studied to detect the more efficient network architecture. The results suggested that Feed forward network trained with training algorithm 'Levenberg-Marquardt' was found to be effective in predicting the monthly groundwater levels. Predicted results using a neural network are in good agreement with values of observed groundwater levels and represent the dynamic characteristics of the given system very well.

REFERENCES

1. ASCE Task Committee on Application of Artificial Neural Networks in Hydrology (2000a), Artificial Neural Networks in hydrology. I: Preliminary Concepts. J. Hydrologic Engineering, ASCE, 5(2), 115-123.
2. ASCE Task Committee on Application of Artificial Neural Networks in Hydrology (2000b), Artificial Neural Networks in hydrology. II. Hydrologic applications, J. Hydrologic Engineering, ASCE, 5(2), 124-137.
3. Belmans, C., J. G. Wesseling, and R. A. Feddes, Simulation of water balance of a cropped soil: SWATRE, J. Hydrol., 63, 271-286, 1983.
4. Bierkens, M.F.P., Modeling water table fluctuations by means of a stochastic differential equation, Water Resource Res., 34(10), 2485-2499, 1998.
5. Box, G.E.P., and G.M. Jenkins, Time Series Analysis: Forecasting and Control, Holden-Day, Boca Raton, Fla., 1976.
6. Coulibaly, P., Anctil, F., Aravena, R., Bobee, B., 2001a. Artificial neural network modeling of water table depth fluctuations. Water Resources Res. 37(4), 885-896.
7. Coulibaly, P., Anctil, F., Bobee, B., 2001b. Multivariate reservoir inflow forecasting using temporal neural networks. J. Hydrol. Eng. 9-10, 367-376.
8. Coulibaly, P., Bobee, B., Anctil, F., 2001c. Improving extreme hydrologic events forecasting using a new criterion for artificial neural network selection. Hydrol. Process. 15, 1533-1536.
9. Feddes, R. A., P. Kabat, P. J. T. Van Bakel, J. J. B. Bronswijk, and J. Halbertsma, Modeling soil water dynamics in the unsaturated zone – State of the art, J. Hydrol., 100, 69-111, 1988.
10. Hipel, K.W., and A.I.McLeod, Time Series Modeling of Water Resources and Environmental Systems, Dev. Water Sci., vol. 45, Elsevier Sci., New York, 1994.
11. Hornik, K., Stinchcombe, M., White, M., 1989. Multilayer feed forward networks are universal approximators. Neural Networks 2, 359-366.
12. Ioannis N.Daliakopoulos, Paulin Coulibaly, Ioannis K.Tsanis, 2004. Groundwater level forecasting using artificial neural network, Journal of Hydrology 309, 229-240.
13. Knotters, M., and J. G. De Gooijer, TARSO modeling of water table depths, Water Resources Res., 35(3), 695-705, 1999.
14. Purna C. Nayak., Satyaji Rao Y.R., Sudheer, K.P., 2005. Groundwater Level Forecasting in a Shallow Aquifer Using Artificial Neural Network Approach. J. Water Resources Management 20, 77-90.

Global Data Assimilation by Artificial Neural Networks for an Atmospheric General Circulation Model: Conventional Observation

Rosangela S. Cintra and Haroldo F. de Campos Velho
Brazilian National Institute of Space Research, , São José dos Campos, São Paulo, Brazil

Abstract - *An Artificial Neural Network (ANN) is designed to investigate its application for data assimilation. This procedure provides an appropriated initial condition to the atmosphere to weather forecasting. Data assimilation is a method to insert observational information into a physical-mathematical model. The goal here is the process for assimilating meteorological observations. The numerical experiment is carried out with global model: the "Simplified Parameterizations, primitivE-Equation DYnamics" (SPEEDY). For the data assimilation scheme, it was applied a supervised ANN: the Multilayer Perceptron (MLP). The MLP-NN is able to emulate the analysis from the Local Ensemble Transform Kalman Filter (LETKF). The ANN was trained with first three months for years 1982, 1983, and 1984 from LETKF. A hindcasting experiment for data assimilation cycle with MLP-NN was performed with the SPEEDY model. The results for analysis with ANN are very close with the results obtained from LETKF.. The simulations show that the major advantage of using MLP-NN is the better computational performance, with similar quality of analysis*.

Keywords: Artificial neural networks, multilayer perceptron, data assimilation, numerical weather forecast

1 Introduction

The procedure which takes atmospheric observed data and creates meteorological fields over some spatial or temporal domain is usually called analysis or data assimilation, when the data are distributed in time and the procedure uses an explicit dynamical model for the time evolution of the atmospheric flow. The fields produced by an analysis or an assimilation must satisfy two basic requirements. On the one hand, they must be close to the observations, at the required spatial and temporal scales. On the other hand, they must verify dynamical and/or statistical relationships which are known to be satisfied by the real atmospheric fields.

In atmospheric modeling the scientist is generally faced with a set of observations of parameters, for instance, wind, temperature, water, ozone, etc., as well as either the knowledge or expectation of correlated behavior between the different parameters. The use of numerical techniques to represent the partial differential equations that represent the model physics is a straightforward way to develop a model. There are many approaches to discretization of the dynamical equations that govern geophysical processes [Randall, 2000].

The numerical weather prediction (NWP) was confronted with having to solve an initial-value problem. Hence, inverse methods were carried over from solid-earth geophysics to estimate, at first, the state of an idealized atmospheric steady-state field. Data assimilation is the objective melding of observed information with model-predicted information. Data assimilation rigorously combines statistical modeling with physical modeling; thus, formally connecting the two approaches. [3] is the standard text on data assimilation. [2] explores the theory of data assimilation and its foundation in estimation theory, and there is a collection of tutorial lectures on data assimilation.

The observational data used in data assimilation are conventional data and satellite data. These data include surface observations and balloon soundings, as well as ship and aircraft observations. Operational satellite data are taken and processed in real-time and distributed around the world. Though small in number, in meteorological data assimilation the conventional data are very important to the quality of the analysis and the forecast.

Atmospheric Data Assimilation is an important task in NWP Centers. Several methods of data assimilation are applied in models of atmospheric and oceanic dynamics. Methods using Artificial Neural Networks (ANN) have been proposed showing consistent results regarding implementation in simple models. This paper presents an approach, which applies an ANN to emulate an Ensemble Kalman Filter (EnKF) as a method of data assimilation in conjunction with an Atmospheric General Circulation Model (AGCM) dynamical nonlinear, using synthetic conventional data simulating balloon soundings.

Data assimilation adds an additional forcing to the representative equations of the physical model; namely, information from the observations. From mathematical point, the assimilating process can be represented by

$$x^a = x^f + W [y^o - H(x^f)] \quad (1)$$

$$W = (H P^f H^T + R). \quad (2)$$

The terms in the equation are as follows: y^o are observations of the constituent, x^f is a model forecast, simulated, estimates of the constituent often called the first guess; H is the observation operator. W is the weighting matrix, generally computed from the covariance matrix of the prediction errors from forecasting and observation. P^f is the error covariance function of the forecast; R is the error covariance function of the observations; xa is the analysis filed with innovation that represents the observation-based correction to the model and the superscript "T" is the matrix transform operation.

The observation operator (H) is a function that maps the parameter to be assimilated onto the spatial and temporal structure of the observations. In its simplest form, the observation operator is an interpolation routine. The error covariance functions, P^f and R, represent the errors, respectively, of the information from the forecast model and the information from the observations. This explicitly shows that data assimilation is the error-weighted combination of information from two primary sources. The analysis is the best estimate of the state of the system based on the optimization criteria and error estimates.

There has been much written in the assimilation literature about the various approaches to the assimilation algorithm and the use of assimilated data sets in many types of applications. However, the computational challenge to the traditional techniques of data assimilation lies in the size of matrices involved in operational NWP models, currently running at a million equations (equivalent to full matrix elements of the order of 10^{12}!). In this scenario the applications of artificial neural networks (ANN) in data assimilation were suggested as a possible technique for data assimilation by [20], [19] and [18]). but the first implementation of the ANN as a approach for data assimilation was employed by [14]. Afterwards, this approach was improved in [13] where the performance of two feed forward (Multilayer Perceptron and Radial Basis Function), and two recurrent (Elman and Jordan) ANN was analyzed. The technique uses neural networks to implement the function:

$$x^a = F_{RNA}(y^o, x^f) \quad (3)$$

where F_{RNA} is the data assimilation process.

Methods using Artificial Neural Networks (ANN) have been proposed showing consistent results regarding implementation in simple models. This paper presents an approach, which applies an ANN to emulate an Ensemble Kalman Filter (EnKF) as a method of data assimilation in conjunction with an Atmospheric General Circulation Model (AGCM) dynamical nonlinear : SPEEDY. The assimilation method chosen for ANN training is LETKF (Local Ensemble Transform Kalman Filter) [15].

2 Methodology

2.1 Data Assimilation technique to training ANN

The analysis is the best estimate of the state of the system based on the optimization criteria and error estimates. The Kalman Filter (KF) is the optimal solution of this estimation problem for linear dynamic systems, see [7], and with linearization problem for nonlinear systems, the FK has been adapted for the Extended Kalman Filter (EKF). Although the methodology of FK has been extremely successful in low-dimensional systems, the direct application to high dimensional systems is problematic due its operations with very large arrays. [10] describes a mathematical introduction about KF and [2] introduces the Kalman filter in the context of atmospheric data assimilation.

Several techniques have been proposed using the EKF and one of them is the Ensemble Kalman Filter (EnKF). The EnKF is a sequential method of data assimilation, proposed by [5] and the first application to an atmospheric system by [16]. It applies an ensemble of model states to represent the model statistical error of the estimate. The scheme of analysis acts directly on the ensemble of model states when observations are assimilated. The ensemble of analysis is obtained by assimilation of perturbed observations for each member of the set of the reference model. Several ways of perturbed observations used to represent the covariance matrix of the analysis, have been derived many schemes from the EnKF approach: the Local Ensemble Transform Kalman Filter (LETKF) is one of them. LETKF was proposed by [17] as an efficient upgrade of LEKF [9. 15]. LETKF separate the entire global grid into independent local patches. The LETKF scheme first separates an entire grid vector into local patch vectors with observations.

The basic idea of LETKF is perform analysis at each grid point simultaneously using the state variables and all observations in the local region centered at that point. Each member of the ensemble gets its forecast: $x_{n,i}^f$ (i=1,2, …, *K*), where *K* is the total members at time t_n. To estimate the state vector of the reference model is used the mean the ensemble forecasts:

$$\bar{x}_n^f = \frac{1}{K}\sum_{i=1}^{K} x_{n,i}^f , \quad (4)$$

considering the model error covariance matrix is:

$$P^a = (K-1)^{-1}\sum_{i=1}^{K}\left(x_{n,i}^f - \bar{x}_n^f\right)\left(x_{n,i}^f - \bar{x}_n^f\right)^T , \quad (5)$$

with the analysis error covariance matrix is:

$$P^a = P^f \left[\left(P^f \right)^{-1} \bar{x}_n^f + H^T R^{-1} H \right]^{-1} \quad (6)$$

and the ensemble analysis is: $x_{n,i}^a$ (i=1,2, …, K) with the its average and error covariance.

LETKF in the local analysis allows different linear combinations of the ensemble members in different regions, and comprehensive analysis explores a larger spatial scale. For local implementation separates groups of neighboring observations to a central point for a region of the model grid.

2.2 Atmospheric Global Circulation Model

The model **S**implified **P**arameterizations primitiv**E**-**E**quation **DY**namics (SPEEDY) is an AGCM (3D model) developing with a core of primitive equations [1, 8]. The model has a simplified set of physical parameterization schemes that are similar to realistic weather forecasting numerical models. (i.e., models used in operational weather forecasting centers).

The model is global with spectral resolution T30L7 (horizontal truncation of 30 numbers of waves and seven vertical levels), corresponding to regular grid with 96 zonal points (longitude), 48 meridian points (latitude) and 7 vertical pressure levels (100, 200, 300, 500, 700, 850, 925 hPa). The package is based on same physical parameterizations adopted in more complex schemes of AGCM like convection (simplified diagram of mass flow), large-scale condensation, clouds, short-wave radiation (two spectral bands), long-wave radiation (four spectral waves), surface fluxes of momentum and energy (aerodynamic formula), and vertical diffusion. Details of the simplified physical parameterization scheme can be found in [12]. The SPEEDY model has spectral dynamics, it is a hydrostatic model in sigma coordinates, and the transformed l vorticity-divergence scheme is described by [1]. The prognostic variables of input and output model are the absolute temperature (*T*), surface pressure (*ps*) component of zonal wind (*u*), component of meridian wind (*v*) and an additional variable and specific humidity (*q*).

The observational data used in data assimilation are conventional data and satellite data. These data include surface observations and balloon soundings, as well as ship and aircraft observations. Operational satellite data are taken and processed in real-time and distributed around the world. Though small in number, in meteorological data assimilation the conventional data are very important to the quality of the analysis and the forecast. The satellite data assures high quality global analyses. It is very clear that assimilation of satellite observations will make a key contribution to that improvement in forecast skills, given the future growth (five orders of magnitude increase in satellite data over ten years) and improvement of the global observing system expected in the area of space-borne observing systems. As a result there is a need an assimilation method able to get the initial field for the numerical model in time to make a prediction. At present most NWP centers cannot assimilate all the data due to computational costs and limitations in storing the data.

The data assimilation experiment is based on synthetic observations simulating a temperature from satellite radiances. The "true nature" assumed to be known and the observations are obtained by adding random noise, simulate observations errors, to model variables in "true" state (integrations are the model without data assimilation).

2.3 Artificial Neural Networks – ANN

ANN is computational system with massively parallel and distributed processing. The ANN has the ability to learn and store experimental knowledge. It is composed of simple processing units (nodes or neurons) that compute certain mathematical functions (usually nonlinear). These units are interconnected by a large number of connections associated with a "weight" that store the knowledge represented in the model and serve to consider the input received by each neuron in the network.

Multilayer Perceptron (MLP) by [4] or [6] is an ANN architecture where the interconnections of inputs to the output layer has at least one intermediate layer of neurons, called hidden layer, It has also, one input vector: $\mathbf{x} = \begin{bmatrix} x_0 & x_1 & \cdots & x_N \end{bmatrix}^T$ and one output vector. $\mathbf{s} = \begin{bmatrix} s_0 & s_1 & \cdots & s_N \end{bmatrix}^T$. A Multilayer network performs a complex mapping $\mathbf{s} = \Psi(\mathbf{w}, \mathbf{x})$ parameterized by the synaptic weights **w**. The set of well-defined procedures to adjust the weights **w** of an ANN to produce a desired output, it is a learning algorithm called back-propagation was used in training the MLP. This algorithm is a supervised process where the network receives input vectors with their corresponding response or desired output. The training process determines the synaptic weights to minimize the error between the output calculated by the network and the expected response to some input vectors. After learning the set of synaptic weights is able to activate the neurons of the MLP and get results for entries that were outside the set of training data, generalizing the information learned.

.

2.4 Experimental settings

The experiment was conducted with the forecast SPEEDY model mentioned and LETKF core modules are applicable to any dynamical model to obtain training data to ANN. The Fortran90 codes (SPEEDY and LETKF) originally developed by [11]. The error covariance at upper levels and surface is

treated in the same way as in [11] to LETKF data assimilation The ANN was developed in Fortran90 codes.

The observational data used in data assimilation are conventional data are those data that are non-satellite data. These data include surface and vertical observations from balloon soundings. Though small in number, in meteorological data assimilation the conventional data are very important to the quality of the analysis and the forecast. The satellite data assures high quality global analyses, and satellite data have been essential of weather forecasts, but firstly we make this new approach with conventional data The observations were generated from "true" model fields, adding random noise with standard deviation (1) to surface pressure, zonal wind component, vertical wind component, temperature and specific humidity values (ps, u, v, t e q}. The variables were located at some grid points from model. The grid points chosen were simulating the conventional network observations of radiosondes, obtaining values for a merged point model (a grid point has observation and a grid point no observation) shown in Figure 1. Both assimilation scheme, LETKF or ANN, have the same numbers of observations at the same grid points.

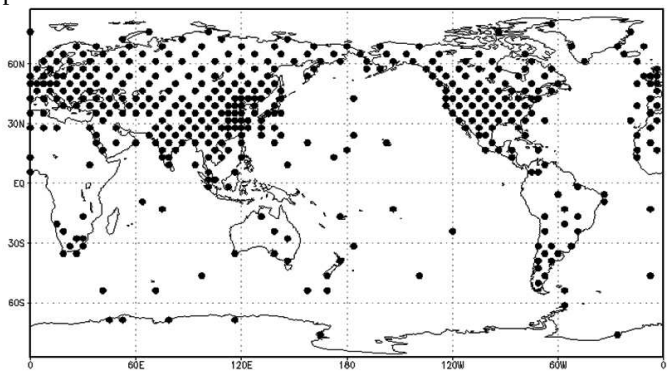


Figure 1 – Realistic distribution of convencional observations, about 451 station for one level.

In this configuration SPEEDY was run for a long time integrations of the state to create "true" to start the integrations of the model. The true integration of the model was made for three years: from 1 January 1982 to 31 January 1985, generating outputs in four times a day (00, 06, 12 and 18 UTC). The LETKF was performed with these *synthetic observations* (about 12,035 points) of the variables to generate the vector analysis and obtain the desired output to train the neural network. The executions of the model with LETKF were made for the periods mentioned to true model. The ensemble of forecasts LETKF has 30 members and the "perturbations of ensemble" consist of random numbers with Gaussian distribution..

In ANN data assimilation scheme we put the local observation influence $\hat{y}^o$ at neighboring grid points with observations point, in zonal and meridional indices, as for the vertical boundaries at the bottom and top levels. We eliminate indices at poles (i.e. no observations at boundaries grid points). This calculation was based on the distance from its neighbor:

$$\hat{y}^o = \sum_{m=1}^{M} y_m^o \left(\frac{1}{r_m^2} \right) \qquad (7)$$

where M is the total neighbors grid points without observations, Δ is a cube-like shape characterized by the horizontal and vertical grid lengths, this is based on the distance at each observation point: $\Delta_r = \left(x_{i,j,k}^f - y_i^o\right)^2 + \left(x_{i,j,k}^f - y_j^o\right)^2 + \left(x_{i,j,k}^f - y_k^o\right)^2$, x^f is grid point without observation that has the forecast variable value, y^o is a grid point with observation; the subscripts (i,j,k) are the indices of localizations points as latitude, longitude, level respectively. Another strategy used to accelerate the processing of MLP training, was to divide the entire globe into six regions. The division was made before LETKF perform for which data were already separated in training areas. The input vectors (observations, forecast value and analysis value at grid points) was collected for two regions in the globe in the vertical (latitude), the northern hemisphere and southern hemisphere, or 90^o to north and 90^o to south; three horizontal regions (longitude), to each hemisphere was divided into three regions of 90^o to each region.

Because of these regions we developed for thirty MLP to all prognostic variable (ps, u, v, T, and q} to each region. These MLP has two inputs (to model and forecast vectors), one neuron in the output (to analysis vector), eleven neurons in a hidden layer, the activation function to ensure nonlinearity of the problem used was the sigmoid tangent hyperbolic.

The network was trained by entering input values of each grid point once, i.e, the analysis is done at grid point where it has observation. The training was made with collected data of the first three months of 1982, 1983 and 1984. The MLP generalization is initiated with ANN data assimilation cycle in the first 00 UTC on 1 January 1985, generating the prediction model and the initial condition of SPEEDY (new global analysis). The MLP generated a hindcasting experiment for data assimilation cycle for the year 1985 with MLP-NN was performed with the SPEEDY model. It was used synthetic conventional observations for January 1985 for both data assimilation system. The LETKF was performed for results comparison.

2.5 Results and conclusions

The results show fields of analysis (q, T, and ps) generated by MLP_NN analysis and fields generated by a realization of LETKF and a realization of MLP_NN for prognostic variables. See Figures of global fields of number 2

to 4, where present the application of MLP_NN as a data assimilation system can reproduce/emulate the LETKF assimilation system. Note in figures (c) the differences between the two analysis field are very small or the results for analysis with ANN are very close with the results obtained from LETKF data assimilation.

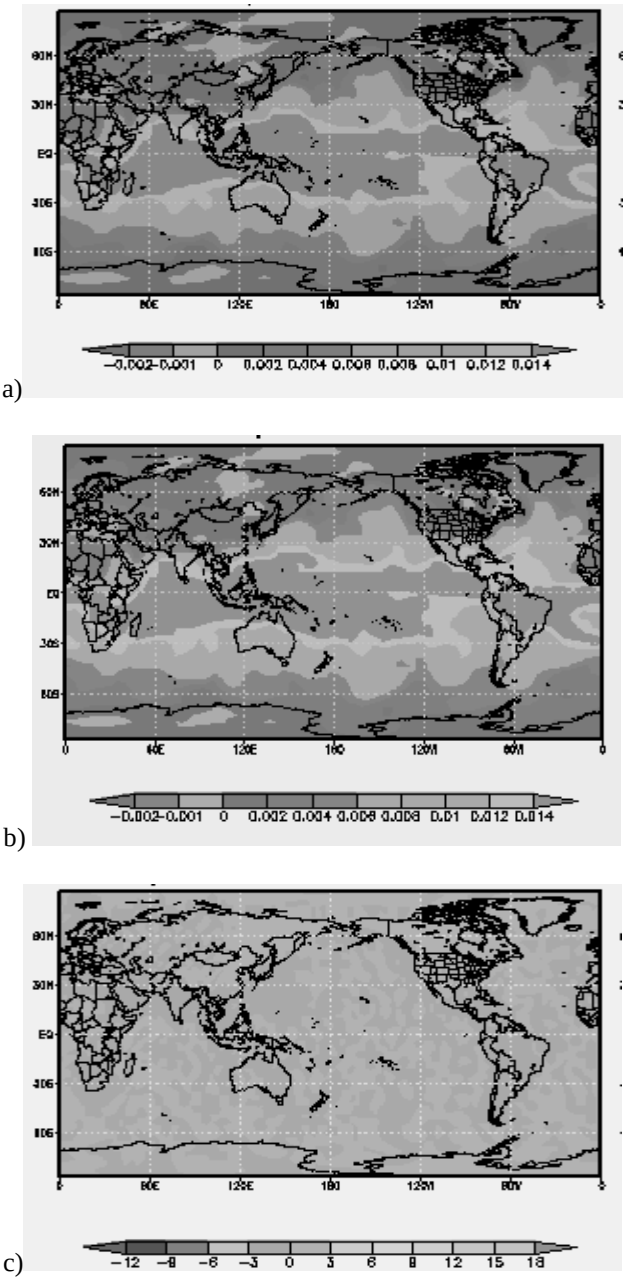


Figure 2 - Analysis field for Humidity (*q*): (a) LETKF and (b) MLP_NN for 1985/01/20-12UTC at level 950hPa. c) Differences from a) and b)

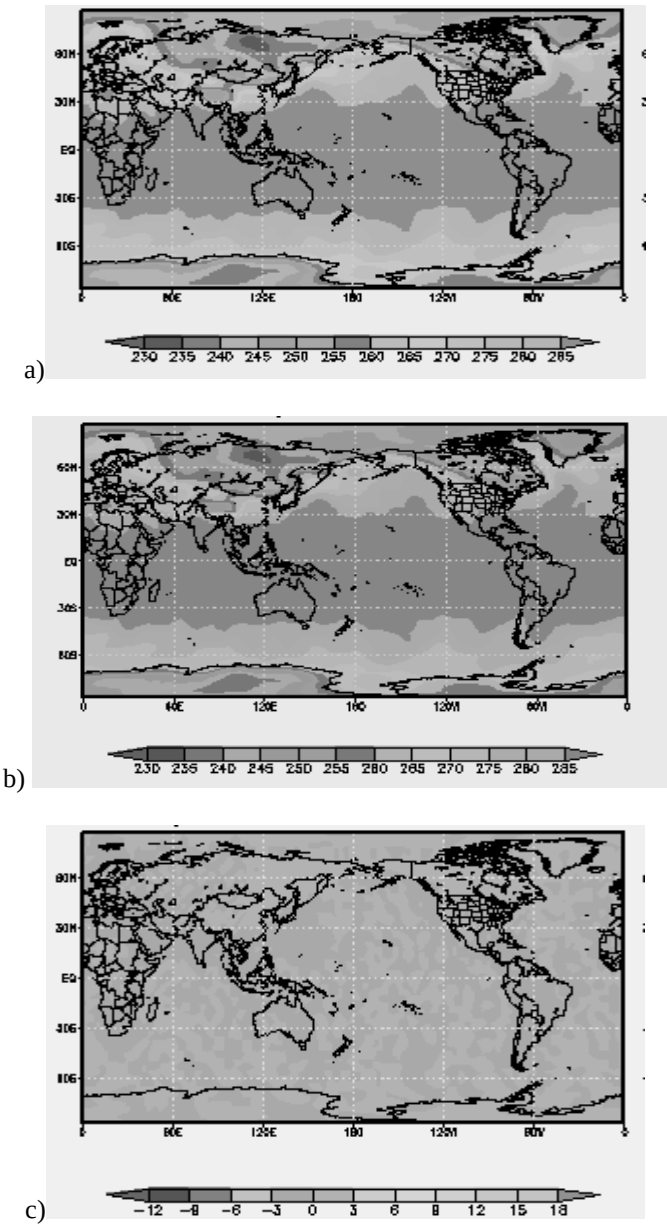
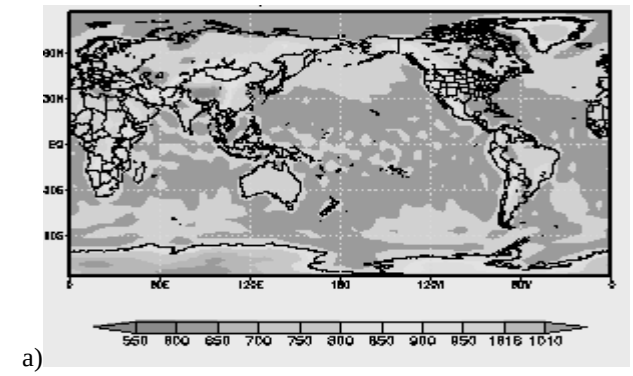
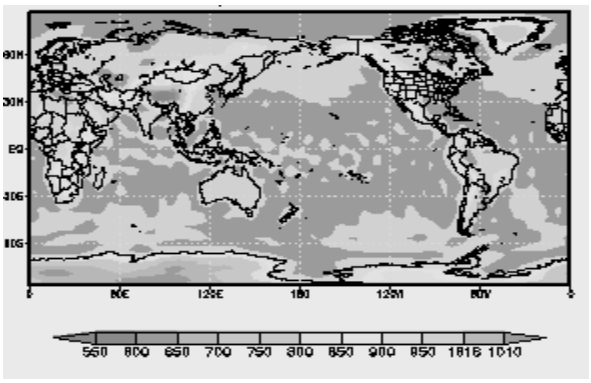


Figure 3 - Analysis field for Temperature (*T*): (a) LETKF and (b) MLP_NN for 1985/01/20 – 12 UTC at level 700hPa. c) Differences from a) and b)



b)



c)

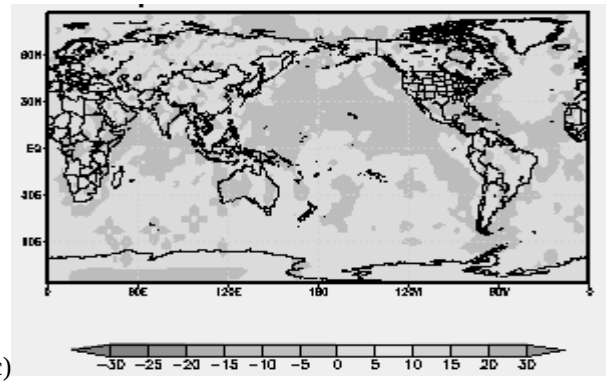


Figure 4 - Analysis field for Surface Pressure (*ps*): (a) LETKF and (b) MLP_NN for 1985/01/11 – 18 UTC.

There are several aspects of the modeling and assimilation problem that stress computational systems and push capability requirements. The common ones in modeling are increased resolution, improved physics, inclusion of new processes, and integration and concurrent execution of Earth-system components that are normally run separately – that is, coupled models. Often, real-time needs define capability requirements. When considering data assimilation the computational requirements become much more challenging. The use of observations from the earth-orbiting satellites in operational numerical prediction models is performed for improving weather forecasts; however, the use of this amount of data increases the computational effort. As a result there is a need an assimilation method able to get the initial field for the numerical model in time to make a prediction. At present most NWP centers cannot assimilate all the data due to computational costs and limitations in storing the data.

The figure 5 shows a cycle of 112 data assimilation, made for 30 days in 2075 with observations of the times (06 and 18 UTC) and 12,035 observations in hours (00, 12 UTC), time was measured in milliseconds, it that the implementation of the assimilation of LETKF These measures show that the computational performance of the RNA tested was higher than the performance of the system LETKF. These results show that the computational efficiency of neural network to the problem of atmospheric data assimilation, 75% less time to run and the same quality in analysis field result.

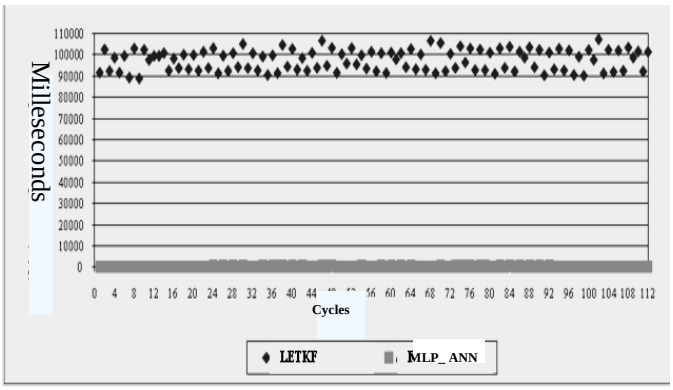


Figure 5 – CPU-time for experiment: MLP-NN (gray color) and LETKF (black color) methods for one month (112 cycles) assimilation.

The CPU-time assimilation with MLP-NN is 75 times faster than LETKF in our numerical experiment. Actually, considering the supervised ANN for data assimilation, the most relevant issue is the computational speed-up for computing the analyzed initial condition.

3 References

Number in square brackets (“[ ]”) should cite references to the literature in the main text. List the cited references in numerical order at the very end of your paper (under the heading `References'). Start each referenced paper on a new line (by its number in square brackets).

[1] W. Bourke. “A multilevel spectral model: formulation and hemispheric integration”. Mon. Wea. Rev., 102, 687-701, 1974.

[2] F. Bouttier and P. Courtier. “Data assimilation concepts and methods. Meteorological Training” Course Series, ECMWF, Reading, UK, 75pp., 1998.

[3] R. Daley, “Atmospheric data analysis”, Cambridge University Press, 1991.

[4] M.W. Gardner and S.R. Dorling, “Artificial neural networks: the multilayer perpectron”, Review of Applic. in the Atmospheric Sciences, Atmospheric Environment, 32(14/15),(1998) 2627-2636.

[5] G. Evensen. “Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics”. J. Geophys. Res.,, 99(C5):10143–10162, 1994.

[6] S. Haykin, “Redes neurais: princípios e práticas”. 2 ed. Porto Alegre: Bookmann, 200.

[7] R. E. Kalman. “A new approach to linear filtering and prediction problems”. Trans. of the ASME–Journal of Basic Engineering, 82(Series D):35–45, 1960.

[8] M. L. Held, L. ,Suarez; "A proposal for the intercomparison of dynamical cores of atmospheric general circulation models". Bull Am Meteorol Soc 75: 1825–1830, 1994

[9] E. Kalnay. "Atmospheric modeling, data assimilation and predictability". Cambridge University Press, New York, 2003.

[10] A. H. Jazswinski,; "Stochastic Processes and Filtering Theory", Academic Press, New York, USA, 1970.

[11] T. Miyoshi, T., "Ensemble Kalman fillter experiments with a primitive-equation global model. 197 p. Thesis (PhD). University of Maryland, College Park, Maryland, USA, 2005.

[12] F. Molteni. "Atmospheric simulations using a gcm withsimplified physical parameterizations. I: model climatology and variability in multi-decadal experiments". Clim. Dyn., 20:175–191, 2003.

[13] F. P. Harter, H.F. De Campos Velho, "Recurrent and Feedforward Neural Networks Trained with Cross Correlation Applied to the Data Assimilation in Chaotic Dynamic", Revista Brasileira de Meteorologia, 20(3), (2005) 411-420.

[14] A. G. Nowosad, "Novas Abordagens para Assimilação de Dados Meteorológicos", tese de Doutorado, Computação Aplicada, CAP-INPE, São José dos Campos (SP), 2001.

[15] E. Ott, B. R. Hunt, I. Szunyogh, A. V. Zimin, E. J. Kostelich, M. Corazza, E. Kalnay, D. J. Patil and J. A. Yorke;"A local ensemble Kalman filter for atmospheric data assimilation", Tellus,, 56(a), 415-428, 2004.

[16] P. L Houtekamer. and H. L. Mitchell, "Data assimilation using an ensemble kalman filter technique". Monthly Weather Review, v. 126, p. 796-811, 1998.

[17] B. Hunt, E. Kalnay, E. J. Kostelich, E. Ott, D. Patil, T. Sauer, I. Szunyogh, J. A. Yorke, A. V. Zemin. "Four-dimensional ensemble kalman filtering". Tellus, v. 56A, p. 273-277, 2004

[18] Y. Tang and W. W. Hsieh; "Coupling neural networks to incomplete dynamical systems via variational data assimilation". Mon. Wea. Rev., v. 129, p. 818-834, 2001.

[19] A. Liaqat, M. Fukuhara, T. Takeda, "Application of neural network collocation method to data assimilation". Comput. Phys. Commun., v. 141, p. 350-364, 2001.

[20] W. W. Hsieh, B. Tang, K. Haines. "A neural network atmospheric model for hybrid coupled modeling". Climate Dynamics, Springer Berlin / Heidelberg, v. 17, Numbers 5-6, p. 445-455, 2001.

A Reconfigurable Neural Network

Alan L. Breitler, Ph.D.
Computer Science Department, University of Maryland University College, Adelphi, Maryland, USA

Abstract – *Allowing a neural network to be trained in the classical way restricts the architectural parameters such as the number of hidden layers and neurons, the intercommunication structure, and the type and values associated with activation functions of the network to predetermined constants. This is tantamount to requiring human learning to be done in a restricted and structured manner, with no opportunity to explore other methods. By allowing network parameters to change as learning progresses, neural network learning has been found to provide results with reduced error.*

Keywords: Reduced error neural networks

1 Introduction

The problems associated with objectively achieving near-optimal results using a neural network focus on the selection of 1) network architecture (i.e., the interconnections between neurons and the number of layers) 2) the number of hidden neurons, and 3) the activation function type (e.g., logistic, step). The selection of the most appropriate combination of these parameters is critical to achieving near optimal results.[1] Having the appropriate architecture but using an insufficient number of hidden neurons, failing to interconnect neurons appropriately, or using an activation function that does not provide the correct sensitivity to inputs may prevent a near optimal solution.

The classical method used to select architectures, numbers of hidden neurons, and activation functions requires some research into the problem itself, and there is no guarantee that the researched solution will be one that provides the best predictor of results based on newly provided (i.e., previously unseen) input data. Experiments have shown that changing the number of neurons applied to a problem can have a significant impact on obtaining results that approach minimum error.[2]

Once such parameter selections have been made, the typical approach is to apply a neural network using these selected parameters consistently and without change. Yet that is not the way in which the human mind operates. Evidence of shifts in neural activity and organization based on the needs of problem solving has emerged [3]. When a problem is presented, the brain functions as a testing ground for various approaches, sifting and sorting responses and at the same time communicating with, organizing and structuring neurons and patterns of neurons until a satisfactory solution is found.

In addition, given that neural networks are a somewhat segmented model of the human brain, replicating brain activity more closely requires the capability to change the structure, connections, and weights given to inputs as the analysis progresses. "To select an appropriate action, we conform to a behavioral rule determined uniquely in each behavioral context. If the rule is not predetermined and must be discovered, we often test hypotheses concerning rules by applying one candidate rule after another."[4] In searching for logical rules, human brains may alternate between probabilistic and rule-based functions, change connectivities, and adopt a flexible approach to the problem at hand. In addition, it has been found that "…functional neuroimaging in humans and electrophysiology in awake monkeys indicate that a fundamental principle of prefrontal function might be adaptive neural coding — in large regions of the prefrontal cortex, neurons adapt their properties to carry specifically information that is relevant to current concerns, producing a dense, distributed representation of related inputs, actions, rewards and other information."[5] There is clearly emerging a body of evidence that the neural structure of the brain solves complex problems using a non-algorithmic approach. If near optimal solutions are to be found with a neural network, restrictions on the network architecture (i.e., the number of layers and coupling parameters between hidden neurons to use a modified form of Hebbian learning [6], as a variation of the ensemble approach [7]), must be relaxed, and different activation functions and inter-neuronal relationships will allow the application of the full power of the network to the pattern identification problem at hand. A modified version of this approach, suggested by Hirose et al [8], added a single hidden neuron when the error level did not drop by a certain percentage. The approach taken

here is more radical, and applies this philosophy to other aspects of the network architecture.

2 Hypothesis

In freeing the neural network from fixed constraints on network architecture, the number of hidden neurons, and the assignment of activation functions, the neural network then mirrors the adaptability of the human brain more closely, and can more effectively find a near optimal solution than can a neural network with fixed architectural and neuronal parameters. Although research has shown that strong constraints operate on the possible universe of architectures [9], the use of evolutionary (i.e., genetic) techniques to gradually retain those characteristics of architectures and neurons that provide low error rates and eliminate those that are associated with higher error rates is a technique that has proven to be successful in enhancing and speeding network training.[10]

3 Procedure

A limited data set is subdivided into two sections, one being the training set and the second being the testing set. The training set data is used to train a network for a limited number of epochs (i.e., iterations through the entire data set). The network thus trained is used to predict values for the testing data set, and if and when the error is reduced from its previous minimum, the characteristics of the new more nearly optimal network are saved and the new error level becomes the minimum error. For the RNN tests, after a certain number of epochs (200,000), the structure, weights, numbers of hidden neurons, and initial weighting factors for all neurons were randomly changed. Using the pre-established error minimum, the newly formed network was then given 200,000 epochs to train. Any new minimums found again resulted in the storage of the network factors, and the process was repeated 1 million times.

4 Data

The training data set is as follows:

Table 1 – Training Data Set

Var 1	Var 2	Result	Var 1	Var 2	Result
2	5	29	8	6	100
9	2	85	9	9	162
10	3	109	6	7	85
10	10	200	7	8	113
3	6	45	1	3	10

The test data set was as follows:

Table 2 – Test Data Set

Var 1	Var 2	Result	Var 1	Var 2	Result
4	5	41	5	3	34
2	1	145	11	11	242
6	6	72	1	1	2
3	7	58	5	9	106
9	3	90	6	5	61

5 Tests

5.1 Fixed parameter tests

To test this hypothesis, a data set of 10 pairs of numbers was used to train a neural network using fixed parameters. The numbers were used to calculate the squares (e.g, if the two numbers were 3 and 5, the sum of the squares was 9 + 25 = 34), and the two squares were added together and provided to the neural network as the true answer for training purposes.

The network was challenged to find the pattern that would allow prediction of the sum of the squares of two numbers. Ten additional pairs of numbers and the sum of the squares for each pair were used as a test set, to avoid having the neural network "memorize" results, a condition known commonly as over-fitting or over-training. This test set consisted of data that the network had not used for training purposes. The network found using the training set was applied to the test set for the calculation of error.

5.2 RNN parameter test

As the second part of the test, a neural network was constructed in such a way that the architecture, number

of hidden neurons, and activation function and initial weights for each neuron were selected at random (after an initial arbitrary assignment) after each attempt to find a solution with less error (i.e., a lower value of the total of all differences between the actual true value and the network predicted value) for the ten pairs of numbers in the test set. The same training and test sets used in the fixed parameter runs were used in this run.

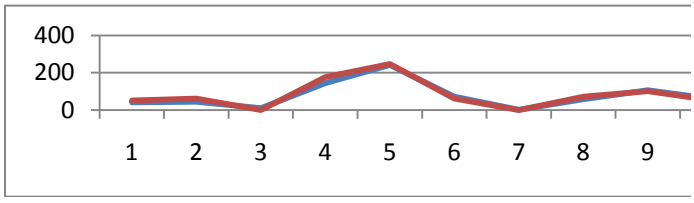
6 Results

The RNN approach proved better than four other approaches tested. Using the same 10 pairs of numbers, the neural network was trained to predict the value of the sum of the squares of the two numbers (i.e., for an input of 2 and 4, the sum of the squares would be 20). The correlation coefficients obtained from the five different approaches were as follows:

System	Correlation Coefficient
Reconfigurable ANN	.984568
Polynomial (GMDH)	.984508
3 Layers Jump	.9651
General Regression	.9518
4 Layers (2 Hidden Layers)	.8134

The graphical representation of the results obtained from the RNN neural network is shown below (red line is the predicted value, blue is the actual value, the horizontal axis is the data set number from 1 to 10, and the vertical axis is the value of the sum of the squares for the data set being tested):

Figure 1 - Results of RNN test run



7 Conclusion

Rather than setting neural network parameters (such as the number of hidden neurons and the type of activation function in each neuron) and keeping them fixed during training, allowing these parameters to adjust as training progresses more closely emulates the activity in the human brain, and converges more rapidly to a solution with less error than conventional training.

While the differences between several of the comparison neural networks were not statistically significant, the fact that equivalent results were obtained in less time indicates that there is a reason for further study of this approach, in addition to the fact that closer emulation of human neural activity is an attainable goal.

8 Future Research

Additional diversity in activation functions and the number of layers of hidden neurons may yield even more rapid convergence to a solution. The variety of different fixed architectures to which the RNN is compared must also be expanded. Cascading of multiple neural networks has been suggested [11], as well as a managed approach wherein the structure of the network itself becomes training data input. The comparison with additional forms of neural networks is also suggested, as there are numerous combinations of numbers of hidden neurons, activation functions, initial weights, and communication interactions between neurons.

A combination of this approach and the typical genetic algorithmic approach (as utilized inside a neural network) is also a subject ripe for further study. The similarity between this approach and a genetic algorithm lies in the random mutation of characteristics. But again, the difference is that typical genetic algorithmic approaches focus on the synaptic weights as opposed to addressing the entire range of neural network characteristics that can be subjected to changes after each "round" of 200,000 epochs.

9 References

[1] Moody, J. (1994), Prediction Risk and Architectural Selection for Neural Networks, From Statistics to Neural Networks: Theory and Pattern Recognition Applications, V. Cherkassky, J.H. Friedman and H. Wechsler (eds.), NATO ASI Series F, Springer-Verlag, 1994.

[2] Obodeh, O. and Ajuwa, C.I. (2009), Evaluation of Artificial Neural Network Performance in Predicting Diesel Engine NOx Emissions, European Journal of Scientific Research, ISSN 1450-216X Vol.33 No.4 (2009), pp.642-653, © EuroJournals Publishing, Inc. 2009 Retrieved February 5, 2011 from http://www.eurojournals.com/ejsr.htm.

[3]Yokoyama, C., et al (2005), A dynamic shift of neural network activity before and after learning-set formation. Cerebral Cortex, 2005 Jun;15(6):796-801. Epub 2004 Sep 15.

[4] Tachibana, K. et al (2009), Neural Activity in the Human Brain Signals Logical Rule Identification, Journal of Neurophysiology, AJP - JN Physiol, September, 2009, vol. 102, no. 3, pp 1526-1537.

[5] Duncan, J. (2001), An Adaptive Coding Model of Neural Function in Prefrontal Cortex, Nature, November 2001, Vol2, pp820-821.

[6] Lauria, F. Prevete, R. and Milo, M (2000), An Adaptable Boolean Neural Network Performing Specific Sequence Learning, IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN'00)-Volume 3 , Issue Date:July 2000 , pp. 3181.

[7] Krasnopolsky, V. (2006), Reducing Uncertainties in Neural Network Jacobians and Improving Accuracy of Neural Network Emulations with NN Ensemble Approaches, 2006 International Joint Conference on Neural Networks, Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada, July 16-21, 2006

[8] Hirose, Y., Yamashita, K. and Hijiya, S. 1991,"Back-propagation Algorithm which varies the number of hidden units". *Neural Networks*, v. 4, Pp. 61-66.

[9] Sole, R. and Valverde, S. (2003), Information Theory of Complex Networks, Universitat Pompeu Fabra (GRIB), Dr Aiguader 80, 08003 Barcelona, Spain and Santa Fe Institute, 1399 Hyde Park Road, Santa Fe NM 87501, USA

[10] Bullinaria, John (2005), Evolved Age Dependent Plasticity Improves Neural Network Performance Fifth International (IEEE) Conference on Hybrid Intelligent Systems (HIS'05), Issued:December 2005 pp. 79-86.

[11] Borenović M.,et al (2009), Cascade-Connected ANN Structures for Indoor WLAN Positioning, Lecture Notes in Computer Science, 2009, Volume 5788/2009, 392-399, DOI: 10.1007/978-3-642-04394-9_48

SESSION

GENETIC ALGORITHMS + SIMULATED ANNEALING

Chair(s)

TBA

Using improved Memetic Algorithm and local search to solve University Course Timetabling Problem (UCTP)

Majid Joudaki
Islamic Azad University, Doroud branch
Doroud, Iran
m.joudaki@gmail.com

Mehdi Imani
Islamic Azad University, Science and Research. Qazvin branch
Qazvin, Iran
m.imani@gmail.com

Niloofar Mazhari
Allameh Dehkhoda College
Qazvin, Iran
nl.mazhari@gmail.com

Abstract- Course Timetabling is a complex problem, happening at the beginning of every semester at universities. In this problem, one of the most important issues is variety of constraints, which results in different ways of timetabling in different universities. Comparing to exact methods which take lots of time to solve UCTP, metaheuristic methods produce a feasible solution within reasonable computation time. In this paper, a hybrid method is presented, which is based on combination of improved Memetic and Simulated Annealing Algorithms. Using Simulated Annealing Algorithm as the local search routine increases exploiting ability of Memetic Algorithm. Also, modifying Crossover operator of Memetic Algorithm and creating initial population by a heuristic-based method improves this algorithm. In order to improve produced chromosomes and decreasing the number of violation of the constraints, a new operator is designed and added to Memetic Algorithm called *improvement operator*. With comparing the results of this method and some modern methods using standard data, efficiency of this method is clear.

Keywords- Memetic Algorithm, Local Search, Simulated Annealing, University Course Timetabling Problem.

I. INTRODUCTION

Timetabling is an issue seen in many aspects of life: working, teaching, transportation, etc. University Course Timetabling Problem (UCTP) is a complicated task, due to educational constraints at universities. According to researches in this field, such constraints are divided into two categories:

Hard constraints: These kinds of constraints must be satisfied in the timetable.

Soft constraints: these kinds of constraints increase quality of the task if they are satisfied, but satisfying these constraints is not compulsory.

Hard constraints have high priorities over soft constraints, and a timetable is considered feasible only if all hard constraints are satisfied in it. For example, a student cannot attend more than one course at the same time. On the other hand, soft constraints are set according to policies of the university and according to need of people using it. For example, a student cannot have three classes in a row in one day. Another point, which is true about any form of UCTP, is that UCTP belongs to NP-hard problems, and we cannot look forward to find an algorithm with time complexity of polynomial order for solving it in general [2]. Solving real timetabling problems using exact algorithms is impossible practically, because for such algorithms, run time increases exponentially when size of the problem increases. Therefore, we can use approximate algorithms to solve them, such as heuristic and Metaheuristic Algorithms. Some examples of tasks using such algorithms are provided in [3, 4, 5, 6, 7, 8, 9, 10].

One of the advantages of the task we have done in this paper is to use heuristic techniques and metaheuristic search methods in considering UCTP. A big difference between this task and previous ones is its flexibility for using real data and the possibility of using it in real university environment. High accuracy, quickness, applying educational policies of university in timetabling, and optimal usage of teacher's and student's time are some advantages of this task.

Memetic Algorithm (MA) [11, 12] is called to a specific class of Metaheuristic Algorithms combining population-based global search (as in Genetic Algorithm) with local search made by each of the individuals [13]. Using exploration ability of evolutionary algorithms and meanwhile combining this advantage with exploiting ability of local search procedures are factors for MA to be successful [13]. While Genetic Algorithm takes inspiration from biological evolution in the nature, MAs mimic cultural evolution in the society [13].

Local search techniques are one choice for solving combinatorial optimization problems [14]. One of the methods of local search is Simulated Annealing (SA). In this method, every neighbor solution which is better than current solution will be accepted. If neighbor solution is worse, it will be accepted by probability of $p = e$. Neighbors of a solution are usually created by applying a simple move on the current solution, and quality of a solution is determined by its fitness (cost function) [15]. In this formula, $\Delta = f(s^*) - f(s)$ and the parameter T represents temperature. Firstly, T has a

high value, and it decreases according to progression formula: $T_{i+1} = T_i * \beta$. In different problems, the cooling rate β and initial temperature are usually different.

The consequence of subjects in this paper is as following: in the next section, one sample of UCTP is introduced for evaluating the algorithm and comparing with other methods. Related works are considered in section III. Proposed algorithm is presented in part IV, and essential pseudo codes are provided for further information. In section V, results of applying the proposed algorithm on some standard data are shown and a comparison to other methods is provided. Conclusion and presenting a proposal for future works are considered in VI.

II. UNIVERSITY COURSE TIMETABLING PROBLEM

A UCTP is generally described as below [17]: a set of events or classes (E), which must be scheduled into 45 timeslots (5 days of 9 timeslots each), a set of rooms (R) where events can take place and a set of students (S), who attend the events, and a set of features (F), which are satisfied by rooms and required by events. Each student attends some of these events and each room has a fixed size. A feasible timetable is one in which every event is assigned a timeslot and a room, and hard constraints are satisfied. These constraints are:

a. No student attends more than one event at the same time.
b. The room is big enough (bigger than or equal to the number of attending students) and satisfies all the features required by the event.
c. Only one event is assigned to each room at any timeslot

Soft constraints are:

a. A student should not have a class in the last timeslot of a day.
b. A student should not have more than two classes in a row.
c. A student should not have a single class on a day.

The problem aims to minimize the number of soft constraints violations in a feasible solution. All infeasible solutions are worthless.

III. RELATED WORKS

In this part some previous works will be under consideration. In these tasks algorithms such as Tabu Search, Genetic Algorithm, Great Deluge, SA, Ant Colony and Memetic are used. Tabu Search Algorithm is one efficient method for solving optimization problems and especially for solving UCTP. Algorithms which make use of Tabu Search are used in [3, 4, 18, 19]. For instance, method [18] continues with locating local search methods between steps of Tabu Search. In [19], effect of two types of neighborhood on Tabu Search in solving UCTP is considered. Similar works are available in [20, 21].

For UCTP, Genetic Algorithm is used in various forms. As an example, the work which is done in [22] is a combination of Genetic Algorithm with a local search method and applying this combination on UCTP. Other methods of timetabling using evolutionary algorithms are reported in [5, 6].

In [23, 24] Great Deluge Algorithm with nonlinear decrease rate is used. In this case, this algorithm has good results, comparing with other methods, but one disadvantage is that we have to set to many parameters in this algorithm, and each parameter has specific value according to the model which is under consideration. Although this algorithm has good answers, due to the problem above cannot be a general algorithm and is highly efficient for some specific models. For a new instance which is out of given instance in this algorithm, we need to reset the parameters. So we do not compare these algorithms with the proposed method.

Optimization algorithms, which make use of simulated annealing, are proposed in [7, 25, 26, 27]. Other examples of multi-purpose techniques which are used in exam timetabling are provided in [8, 28] and with details in [29]. In [30, 31] two algorithms are presented in the basis of Ant Colony Algorithm, for solving UCTP. In [9, 10, 13, 17, 32, 33] MA is applied to UCTP.

IV. THE PROPOSED METHOD (MA_SA_UCTP)

As mentioned in section 1, MA is a powerful algorithm for solving optimization problems. MA_SA_UCTP acts as follows. Each chromosome is set using heuristic initialization procedure. Then the local search procedure (which is based on SA) is recalled to improve the chromosome. In the next step, we apply improvement procedure on the chromosome to improve it as much as possible. In a while loop, two chromosomes of population are selected due to their fitness. Crossover and Mutation operators are applied with probability of pC and pM respectively and a child is produced. Then SA and improvement procedures are recalled and improve the child. Finally, this child is

compared to the worst member of population and if the child is better than that member, it will be replaced with the member. This procedure will continue until meeting stopping rules. Figure 1 illustrates the algorithm.

input : a problem instance **I**
foreach chromosome pop_i in pupolation **do**
  $pop_i \leftarrow$ heuristic initial chromosome
  $pop_i \leftarrow$ apply Local Search(SA) on pop_i
  $pop_i \leftarrow$ apply improvement routine on pop_i
end for
sort pupolation based on fitness
while termination condition not reached **do**
  select two parents from population by proportional selection
  $child \leftarrow$ child solution after crossover with a probability pC
  $child \leftarrow$ child solution after mutation with a probability pM
  $child \leftarrow$ child solution after applying Local Search(SA)
  $child \leftarrow$ child solution after applying improvement
  if $child$ is better than worst member of the population
    $worst\ member \leftarrow child$
  sort pupolation based on fitness
end while
  output : The best solution pop_0 achived for **I**

Fig. 1: MA pseudo code presented for UCTP

A. Presenting solution using Chromosome

Figure 2 illustrates the structure of a chromosome. In this presentation, every gene consists of a triple <event, timeslot, room>. Chromosomes are equal in length and this length is the number of events. This presentation is simple and it is easy to apply MA operators on it.

event_1	.....	event_i	.....	event_n
timeslot_1	.....	timeslot_j	.....	timeslot_t
room_1	.....	room_k	.....	room_r

Fig. 2: the structure of a chromosome

B. Heuristic Initialization Procedure

UCTP is similar to the graph coloring problem [2], so we can use heuristic procedures of graph coloring to produce chromosomes [36]. By considering each event as a vertex, Events cannot take place in the same timeslots, so we add edges (limitation) between each pair of vertices. In this manner, we can convert lots of simple timetabling problems to graph coloring problems and vice versa. We can consider every timeslot as a color. Figure 3 presents heuristic procedures in initialization. This step begins with an empty timetable and a list of events (L) which consists of unallocated events (firstly |L|=n). Using heuristic procedures in figure 3, each time one event and one appropriate location are selected and the event is allocated to that location.

h_1	Selecting an event with lower appropriate location in timetable
h_2	Selecting an event with higher conflict with other events
h_3	Selecting an event in a random manner
h_4	Selecting an appropriate location for lower unallocated events
h_5	Selecting a location with lower number of events in a time slice
h_6	Selecting a location in a random manner
h_7	Selecting an event with lower number of students

Fig 3: heuristic procedures to produce chromosomes

C. Local Search Procedure

This procedure is based on SA. The code is presented in figure 4.

input : a solution instance $\boldsymbol{s}$
$iteration \leftarrow 0$
while $T > tMin$ **do**
  for $iteration = 1$ to $maxIteration$ **do**
    apply $neighborhoodSearch$ to the $\boldsymbol{s}$
    $\boldsymbol{s}^* \leftarrow$ select a neighbour of $\boldsymbol{s}$ randomly
    $\Delta \leftarrow f(\boldsymbol{s}^*) - f(\boldsymbol{s})$
    if $\left(\Delta <= 0 \ \textbf{OR}\ e^{-\frac{\Delta}{T}} < rand[0,1]\right)$ **then**
      $\boldsymbol{s} = \boldsymbol{s}^*$
    increament $iteration$ by 1
  end while
  $T \leftarrow$ Apply cooling rate (β) to the T
end while
  output : Solution **s** after applying SA

Fig. 4: SA procedure used as local search in MA

In this procedure, three different neighborhood structures are used: 1) Event movement from one timeslot to another. 2) Exchanging two events timeslots with each other. 3) Exchanging three events timeslots with each other in a circular manner. We should mention that events are selected randomly in all above structures.

D. Crossover Operator

This operator selects genes as follows: firstly, one event is selected and the number of violation of the soft constraints is studied in both parents and the gene for this event is selected from a parent with lower soft constraints violations. This gene is transmitted to the child with its related genes (genes which consist of events having the same student with this event). This procedure continues until all genes are transmitted. Whenever we have parents with same situation, one parent is selected randomly for selection of the gene. Figure 5 shows this procedure.

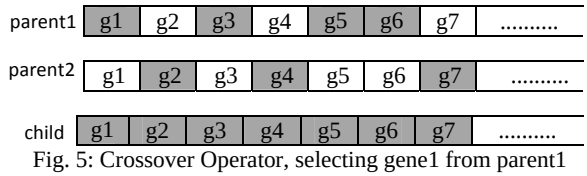


Fig. 5: Crossover Operator, selecting gene1 from parent1 and transmitting this gene to the child with its related genes. This is true for gene2 from parent2.

E. Improvement of Chromosome

This operator decreases the number of soft constraints violations by moving the students from one course group to another one. A course group consists of a course which is presented in different times. If we can decrease the number of soft constraints violations by moving one student from one course to another, this movement is done.

V. EXPERIMENTAL RESULTS

MA_SA_UTCP is coded with Java and evaluated on a PC with a memory of 512 MB and Intel 2.0 GHz processor. MA_SA_UTCP parameters are presented in table 1.

Value	Parameter
100	Number of generation
40	Amount of population
0/8	Crossover probability
0/07	Mutation probability
200	Initial temperature
0/005	Final temperature
0/999	Temperature decrease rate

Table 1: Parameters used in MA_SA_UCTP

MA_SA_UTCP is applied on [34] data. Although this data does not show all aspects and limitations of a real problem [35], but it is a good standard for comparing our proposed method to other applied methods for this data.

In table 2, MA_SA_UTCP is compared to some other new methods. First column presents instances of the problem. In this column letters *s*, *m* and *l* represent small, medium and large. Second column shows MA_SA_UTCP results for different instances. In each column, numbers show the number of soft constraints violations. Reference numbers of compared methods are provided above each column. In each row, the lower (or the best) values are in bold. 'Inf' expressions are used for methods which are not able to find a feasible solution in a reasonable time (this time differs from one method to another and is between 3 to 10 hours).

Instances	MA_SA_UCTP	[5]	[6]	[10]	[13]	[22]	[36]	[37]
s1	**0**	**0**	**0**	**0**	**0**	2	5	**0**
s2	**0**	**0**	**0**	**0**	**0**	4	3	**0**
s3	**0**	**0**	**0**	**0**	**0**	2	2	**0**
s4	**0**	**0**	**0**	**0**	**0**	**0**	3	**0**
s5	**0**	**0**	**0**	**0**	**0**	4	1	**0**
m1	**144**	317	242	221	227	254	316	236
m2	**142**	313	161	147	180	258	243	158
m3	**207**	357	265	246	235	251	255	261
m4	**126**	247	181	165	142	321	235	176
m5	169	292	151	**130**	200	276	215	147
l	942	inf	inf	529	inf	1027	inf	**296**

Table 2: Comparing MA_SA_UCTP with some existing methods. inf is unacceptable.

For small, medium and large instances run time of the proposed algorithm is 60, 300 and 600 minutes, respectively. Results show that MA_SA_UTCP is able to find a feasible solution for every sample. Figure 6 represents improvement percentages of proposed method and methods in [10, 37]. These two methods are better than others and are chosen for comparison. As in bar chart, our proposed method acts better for all samples (it has positive percentage) except for m5. For instance m5 the percentage is negative and in this case other two methods act better.

If the solutions of MA_SA_UTCP and compared methods are s and s', for one instance, then improvement percentage is calculated as follows:

$$ImpPercentage = \frac{\left(f(s) - f(s')\right) * 100}{f(s')}$$

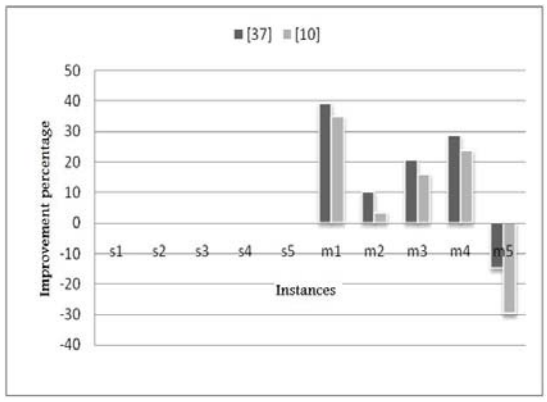


Fig. 6: comparing MA_SA_UTCP with methods in [10, 37]. Horizontal line represents test samples and vertical one represents improvement percentage.

Highest improvement amount is seen in instance m1. In this case, we have an improvement of 34.8 times higher comparing to [10] and an improvement of 38.9 higher comparing to [37].

VI. CONCLUSION AND FUTURE WORKS

In this paper, we studied MA and a combination of MA and local search procedure which is based on SA to solve UCTP. Using heuristic initialization procedure was helpful for satisfying hard constraints and making initial solutions acceptable. MA_SA_UCTP was applied on real data at the computer and electrical department of Isfahan University. According to the results, this method is comparable to experts' paper works and it needs less time to produce timetable. We can make use of this algorithm, in the case that we formulate UCTP in this department and solve some exception cases. In future works we can develop new operators for MA (such as improvement operator in this paper). Another alternative is to change standard operators of MA. Applying heuristic methods to satisfy soft constraints can be considered in solving this problem.

VII. REFERENCES

[1] E. Burke, J. Kingston, K. Jackson, and R. Weare. "Automated University Timetabling: The State of the Art" *The Computer Journal*, Vol. 40, No. 9, pp. 565-571, 1997.

[2] R Lewis. "A survey of metaheuristic-based techniques for university timetabling problems", *OR Spectrum*, Vol. 30, No. 1, pp. 167-190, 2008.

[3] D. Costa. "A tabu search algorithm for computing an operational timetable" *Eur J Oper Res*, Vol. 79, pp. 98–110, 1994.

[4] L. Gislen, C. Peterson, and B. Soderberg. "Teachers and classes with neural networks" *Int J Neural Syst*, Vol. 1, pp. 167–176, 1989.

[5] D. Corne, P. Ross, and H. Fang. "Evolving timetables" In: Lance C. Chambers (ed) *The practical handbook of genetic algorithms*, Vol. 1, CRC, Florida, pp. 219–276, 1995.

[6] M.Carrasco, and M. Pato. "A multiobjective genetic algorithm for the class/teacher timetabling problem," *In: Burke E, Erben W (eds) Practice and theory of automated timetabling (PATAT) III*, Vol .2079, Springer, Berlin, pp. 3–17, 2001.

[7] D. Abramson, H. Krishnamoorthy, and H. Dang. "Simulated annealing cooling schedules for the school timetabling problem" *Asia Pacific J Operational Res*, Vol. 16, pp. 1–22, 1996.

[8] E. Burke, Y. Bykov, and M. Petrovic. "A multicriteria approach to examination timetabling," *In: Burke E, Erben E (eds) practice and theory of automated timetabling (PATAT) III*, Vol. 2070, Springer, Berlin, pp. 118–131, 2001.

[9] A. Alkan and E. Ozcan." Memetic algorithms for timetabling evolutionary computation," *Proc of the 2003 IEEE Congress on Evol Comput*., Vol. 3, pp. 1796–1802, 2003.

[10] S. Abdullah, E. K. Burke, and B. Mccollum. "A hybrid evolutionary approach to the university course timetabling problem," *In Proceedings of the IEEE Congress on Evolutionary Computation*, Singapore, September 2007.

[11] P. Moscato. "On evolution, search, optimization, genetic algorithms and martial art: towards memetic algorithms," caltech concurrent computation program, Technical Report, 1989.

[12] P. Moscato. "Memetic algorithms: A short introduction," *In: New Ideas in Optimisation, Mcgraw-Hill'S Advanced Topics In Computer Science Series*, pp. 219–234.

[13] O. Rossi-Doria, B. Paechter. "A memetic algorithm for university course timetabling" in: *Combinatorial Optimisation 2004 Book of Abstracts*, Lancaster, UK, Lancaster University, 2004.

[14] J. Holland, *Adaptation in Natural and Artificial Systems*, Ann Arbor, MI: University of Michigan Press, 1975.

[15] E. Burke, Y. Bykov, J. Newall, and S. Petrovi. "A Time-Predefined approach to Course imetabling" *Journal of Operations Research*, Vol. 13, No. 2 ,pp. 139-151, 2003.

[16] M . Nandhini, and D. S. Kanmani. "A Survey of Simulated Annealing Methodology for University Course Timetabling" *International Journal of Recent Trends in Engineering*, Vol. 1, No. 2, pp. 255-257 May 2009.

[17] M. Chiarandini, K. Socha, M. Birattari, and O. Rossi-Doria, "An effective hybrid approach for the university course timetabling problem," FG Intellektik, FB Informatik, TU Darmstadt, Germany, Tech. Rep AIDA-2003-05, 2003.

[18] A. Schaerf. "Tabu search techniques for large high-school timetabling problems," *In: Proceedings of the thirteenth national conference on artificial intelligence*. AAAI Press/MIT Press, Portland (OR), pp. 363–368, 1996.

[19] C. H. Aladag, G. Hocaoglu, and M. A. Basaran, "The effect of neighborhood structures on tabu search algorithm in solving course timetabling problem" *Expert Systems with Applications*, doi: 10.1016/j.eswa.2009.04.051, 2009.

[20] C. H. Aladag, and G. Hocaoglu. "A tabu search algorithm to solve course timetabling problem" *Hacettepe Journal of Mathematics and Statistics*, Vol. 36, NO.1, pp. 53–64, 2007.

[21] R. Alvarez, E. Crespo, and J. M. Tamarit. "Design and implementation of a course scheduling system using tabu search" *European Journal of Operational Research*, Vol. 137, pp. 512–523, 2002.

[22] S. Abdullah, and H. Turabieh. "Generating University Course Timetabling Using Genetic Algorithms and Local Search," *International Conference on Convergence and Hybrid Information Technolog*, 2008.

[23] D. Landa-Silva, and J. H. Obit, "Great Deluge with Nonlinear Decay Rate for Solving Course Timetabling Problems," *Proceedings of the 2008 IEEE Conference on Intelligent Systems (IS 2008)*, IEEE Press, pp. 8.11-8.18, 2008.

[24] Dario Landa-Silva, Joe Henry Obit. "Evolutionary Non-Linear Great Deluge for University Course Timetabling," *Proceedings of the 2009 International Conference on Hybrid Artificial Intelligence Systems (HAIS 2009)*, Lecture Notes in Artificial Intelligence, Vol. 5572, Springer, pp. 269-276, 2009.

[25] D. Abramson, "Constructing school timetables using simulated annealing: sequential and parallel algorithms" *Manag Sci*, Vol. 37, No. 1, pp. 98–113, 1991.

[26] F. Melicio, and J. Caldeira. "Timetabling implementation aspects by simulated annealing," *In: Jifa Gu (ed), IEEE systems science and systems engineering,* Beijing, Aceite, pp. 553–557, 1998.

[27] S.Elmohamed, G. Fox, and P. Coddington. "A comparison of annealing techniques for academic course scheduling," *In: Burke E, CarterM(eds) Practice and theory of automated timetabling (PATAT) II*, Vol. 1408, Springer, Berlin, pp. 146–166, 1998.

[28] S.Petrovic, and Y. Bykov. "A multiobjective optimisation approach for exam timetabling based on trajectories," *In: Burke E, De Causmaecker P (eds) The paractice and theory of automated timetabling (PATAT) IV,* Vol. 2740, Springer, Berlin, pp. 181–194, 2003.

[29] L. Paquete, and C. Fonseca. "A study of examination timetabling with multiobjective evolutionary algorithms," *In:4th metaheuristics international conference (MIC 2001)*, Porto, pp. 149–154, 2001.
[30] K.Socha, and M. Samples. "Ant algorithms for the university course timetabling problem with egard to the state-of-the-art," *In: Evolutionary computation in combinatorial optimization (EvoCOP 2003)*, Vol. 2611, Springer, Berlin, pp. 334–345, 2003.
[31] K. Socha, J. Knowles, and M. Samples. "A max-min ant system for the university course timetabling problem," *Proceedings of the 3rd International Workshop on Ant Algorithms (ANTS 2002),* Springer Lecture Notes in Computer Science, Vol. 2463, pp. 1-13, 2002.
[32] B. Paechter, A. Cumming, M. G. Norman, and H. Luchian. "Extensions to a memetic timetabling system," *Proceedings of the 1st International Conference on Practice and Theory of Automated Timetabling,* LNCS 1153, pp. 251–265, 1996.
[33] O. Rossi-Doria, M. Sampels, M. Birattari, M. Chiarandini, M. Dorigo, L. Gambardella, J. Knowles, M. Manfrin, M. Mastrolilli, B. Paechter, L. Paquete, and T. Stutzle."A comparison of the performance of different metaheuristics on the timetabling problem," *Lecture Notes in Computer Science* 2740, pp. 329–351, 2002.
[34] http://iridia.ulb.ac.be/supp/IridiaSupp2002-001/index.html
[35] B. McCollum. "University Timetabling: Bridging the Gap between Research and Practice," *Proc of the 6th Int Conf on the Practice and Theory of Automated Timetabling*, pp. 15–35, 2006.
[36] M. A. Al-Betar, A. T. Khader, and T. A. Gani. "A Harmony Search Algorithm for University Course Timetabling," *In:E. Burke,M. Gendreau. (eds.). The Proceedings of the 7th International Conference on the Practice and Theory of Automated Timetabling,* Montréal, Canada, 2008.
[37] S. Abdulla, and A. R. Hamdan. "A hybrid approach for University Course Time Tabling" *IJCSNS*, Vol.8, No.8, 2008.
[38] D. J. A. Welsh, and M. B. Powell. "An upper bound for the chromatic number of a graph and it's application to timetabling problems". *Computer Journal*, Vol. 10, No. 1, pp. 85-86, 1967.

Solving a Public Sector Sustainable Supply Chain Problem: A Genetic Algorithm Approach

Ernesto Del R. Santibanez-Gonzalez[1], Geraldo Robson Mateus[2], and Henrique Pacca Luna[3]
[1]Department of Computer Science, Federal University of Ouro Preto, Ouro Preto, MG, Brazil
[2]Department of Computer Science, Federal University of Minas Gerais, Belo Horizonte, MG, Brazil
[3]Computer Science Institute, Federal University of Alagoas, Maceió, AL, Brazil

Abstract - *This paper addresses a sustainable supply chain network design problem that arises in the public sector. There is little research being done in mathematical modeling and solutions methods for these problems. The paper describes a mixed-integer 0-1 model (MIP) for solving this sustainable problem in which we have to determine a fixed number of facilities to be located at sites chosen from among a given set of candidate sites. Sustainable issues are integrated into the model by reducing the greenhouse gas emissions produced by the transportation and the operation of the facilities. We propose a hybrid genetic algorithm (GA) for solving this problem by introducing a greedy-like procedure during the feasibility phase. In order to validate our GA solutions we used GAMS to obtain optimal objective values on the MIP. We report computational results for instances generated from a known OR test library.*

Keywords: Sustainable supply chain, green supply chain, optimization, genetic algorithms

1 Introduction

In 1987 the United Nations World Commission on Environment and Development (UNWCED) published Our Common Future Report. In this report was defined sustainable development as "development that meets the needs of the present without compromising the ability of future generations to meet their own needs". This report was a kick-off for a number of interdisciplinary studies in the field of sustainability.

By 2020 the economic impact of climate change in the world will reach 20% of the global GDP. Considering five countries with the highest GNP, this impact means entire losses of about 6,000 billion USD every year, or losses of 2,700 billion USD every year only for EE.UU. The man-made greenhouse gas emissions (GHG) are one of the main causes of the climate change. Under business as use scenarios, it is expected that the GHG emissions should increase of 40 GtCO2e (dioxide carbon equivalent) emitted per year in 2002 up to almost 53 GtCO2e annually about 2020.

Considering the magnitude of the impact on the global economy, the environment and the society, many agencies of the government and international institutions are taking actions to reduce and to control the emission of GHG. The Kyoto Agreement, signed by 35 countries, was one of the pioneering intentions of reducing the emission GHG. This agreement placed the target to reduce about 2012 the global emission of coal for an average of 5.4 % in relation to the levels of 1990. In this respect, this paper focuses on how governments can deploy supply chain networks for servicing people while minimizing the costs of installation, operation and transportation and, at the same time reducing the GHG emissions.

There are principally two ways that the human mind contributes to the emission of GHG: production of energy (generation of energy) and transport and logistics, apart from the deforestation. Both are basic activities that every company as every government agency realizes to satisfy end costumers.

Governmental agencies and companies adopting a friendly sustainable management are facing a number of changes, from the strategy level till the operational point of view, affecting their people and impacting their business processes and their technology. In this regard, as pointed out in [1], "the strategic level deals with decisions that have a long-lasting effect on the firm. These include decisions regarding the number, location and capacities of warehouses and manufacturing plants, or the flow of material through the logistics network". They established a clear link between facility location models and strategic decisions of Supply chain management (SCM). Also, governmental agencies and companies realized that to be committed with sustainable practices could imply changes in the criteria to design and to manage supply chain. That is to say, in addition to the costs of transport, operation and installation and considerations on the level of service, the sustainable models need to consider GHG emission costs.

Supply chain design based on economic consideration has been well covered in the literature. On the other hand, the field of sustainable supply chain design and management (SSCM) is quite new [2]. The greatest benefits of applying SCM are obtained by an extended analysis including organizations upstream -closer to the raw materials- and downstream -closer to the consumer- of the supply chain and then back again so that the unsold products are recycled. But, by extending the focus, what this really does implies more organizations, multiplying the relation between the organizations and getting a more complex supply chain (SC) to manage. Then the focus of the supply chain management literature has been on dyadic networks (supplier units-customer units) as we do in this paper [3].

This paper proposes a genetic algorithm for solving a supply chain network design problem that arises in the public sector considering sustainable constraints in the form of restrictions on the dioxide carbon equivalent emissions. The authors are not conscious of any article tackling the problem of sustainability that arises in the location of such public facilities as schools or hospitals. We present a mixed-integer 0-1 facility location model who allows to analyze the impact of restrictions in the GHG emissions on the fixed and transportation costs and in the location of facilities.
In this paper, in Section 2 is analyzed some literature in connection with the problem. In section 3 is presented the mixed-integer 0-1 programming model. In section 4, we discuss the genetic algorithm implementation for solving the problem. In section 5 we provide some numerical results. Finally in section 6 we give some conclusions of the work.

2 Literature review

The UNWCED report published in 1987 defined sustainable development as "development that meets the needs of the present without compromising the ability of future generations to meet their own needs".
Above and beyond other aspects, there are principally two kinds of questions related to the sustainability. One of them is the emission of GHG. For the production of a big quantity of GHG to satisfy the present needs – for example, in the processes of manufacture and distribution - we affect the climate of a dangerous way that finally is going to compromise the capacity so that future generations could satisfy their own needs. On the other hand, waste production is another form of affecting the environment. In this paper we focus on reducing the GHG emissions caused by the operation of pubic facilities and the transportation activities to satisfy the demand. We model the problem of locating facilities in order to satisfy a fixed demand, minimizing installation and operating costs and constraining the GHG emissions.
Supply chain management (SCM) is being used to address the problem of reducing the economic impact of climate change generated by GHG emissions. For the purpose of this paper, SCM is a multidisciplinary management approach to master a set of interacting organizations. These organizations share different resources, products, services and information, with the target to obtain competitive advantages and to improve the profitability, both in the individual form and in the collective form [1]. According to some authors [3], the focus of the literature of SC has been in networks of dyadic (supplier firms – customer firms) as we do in this work.
Disciplines integrating environmental practices into the supply chain have been called in a number of different ways. Some of them are: Sustainable Supply Chain Management (SSCM) and, Green Supply Chain Management (GSCM). In [4] did a careful review of the literature and he showed that a wide frame of reference for GSCM has not been sufficiently developed. As a consequence, the author identifies the need "of a succinct classification to help academicians, researchers and practitioners in understanding integrated GSCM from a wider perspective". Finally, he defined GSCM as an integrated environment, including product design, sourcing and selection of material, manufacturing processes, delivery of the final product to the consumers, and end-of-life management of the product after its useful life. In this paper, we do not make any distinction between sustainable and green supply chain.
Mathematical modeling for designing sustainable supply chain is attracting many researchers to this field. But, according to some authors [2], the field of sustainable supply chain design and management (SSCM) is quite new. In this work, the proposed model focuses on two sustainable issues: economic and environmental aspects of GHG emissions. On the other side, till now much research has been done in the field of private companies' location theory. The authors are not aware of any paper addressing the sustainable problem that arises in the location of such public facilities as schools or hospitals.
Reference [5] developed a multi-objective mixed-integer 0-1 model for deciding location and capacity expansion of facilities (plants), and transportation issues in a given planning horizon. They maximize profit and minimize the environmental impact of the plant operations while satisfying the market demand for products. They presented numerical results for a small problem of 3 candidate plants, 3 customers, 2 products, 2 raw materials and 5 periods. In a later work [6], they extended the previous model and reformulate the problem as a stochastic programming model that can address the decision-making process under uncertainty. Authors in [7] proposed a mixed-integer 0-1 programming model for the GSC design problem. Taking into account environmental aspects, they analyze the impact of transportation, subcontracting, and production activities on the design of a supply chain network. The problem considers a three-echelon multi-products supply chain network formed by a set of suppliers, a set of subcontractors (plants), and a set of costumer zones. The model integrates into the objective function the total amount of GHG emissions produced by transportation and production activities, and determines the equivalent carbon credits generated for different configurations of the supply chain. The model is tested considering the case of a steel product manufacturer with three freight transportation modes, a product with two semi finished products that are manufactured from four parts, and at least two suppliers are competing to supply each part. The model is first solved by CPLEX Interactive Optimizer V10.0. The authors also use Goal Programming to determine the best trade-offs between two conflicting objectives: the total logistics cost and carbon emissions. In [8] is extended the previous model considering life cycle assessment (LCA) principles in addition to the traditional material balance constraints at each node in the supply chain. They propose a multi-objective mixed-integer 0-1 model to support sustainable supply chain design over a long-term period of time. The model distinguishes between solid and liquid wastes, as well as gaseous emissions due to various production processes and transportation modes. The model is used to evaluate the tradeoffs between economic and environmental objectives under various cost and operating

strategies for an aluminum company. Finally, [9] used a mixed-integer 0-1 programming model including carbon emissions restrictions for designing green supply chains. The problem is to decide which plants and Distribution Centers (DCs) to open, how the DCs are allocated to the plants, and how the DCs distribute multiple types of products to satisfy retailers' demands. The objective is to minimize the total facility opening and products distribution costs subject to the total carbon emission is not more than a predetermined emission cap. They formulate the problem as a two-echelon multi-commodity facility location problem with a carbon emission constraint. They present numerical results for a 7 candidate plants, 18 candidate DCs, 63 retailers, and a single type of product. To solve the instances, the authors use ILOG CPLEX 11.0 MIP solver in the GAMS modeling language.
The problem of locating facilities and allocating customers is not new to the operations research community and covers the key aspects of supply chain design [10]. In [1] authors established a clear link between location models and strategic SCM. While authors in [11] pointed out that this problem is one of "the most comprehensive strategic decision problems that need to be optimized for long-term efficient operation of the whole supply chain". Notice that some small changes to classical facility location models are quite hard to solve [12]. For example, in [13] a genetic algorithm was used to solve a kind of facility location problem on test networks with 10 potential facility sites and 30 demand points. In this paper we focus on the sustainable supply chain design problem that arises in governmental agencies, where you have to decide the location of schools, hospitals, police stations, fire stations, and so on, taking into account sustainable issues.

3 Problem formulation

The sustainable supply chain network design problem consists in deciding the number and location of facilities, and the allocation of customers to these facilities, minimizing the installation and transportation costs integrated with GHG emissions constraints. We consider a public supply chain that provides products/services and it consists of two layers hierarchically related. In the general case, these supply chain consists of two layers (mid and high) of distinct types of facility. For example, health care systems may consist of local clinics and hospitals or medical centers; higher education systems may consist of technical schools and universities. For further details on related problems see the paper by [14]. Our problem is uncapacitated by nature, following most of the research on locating public facilities, i.e., we do not restrict the capacity of the facilities to service the demand. Our interest is to analyze how service costs of governmental agencies located in both layers will be affected by sustainable restrictions. We assume that GHG emissions come mainly from the operation of the facilities located in both layers and from the transportation activities involved to service a fixed demand. Notice that, the clients can be attended by only one facility of the mid layer and each mid layer facility must be allocated to one high layer facility. We suppose that GHG emissions are proportional to the demand, i.e. population to be attended, and the travel distance. Then each facility has a carbon footprint proportional to the demand attended. We model this using a parameter to be adjusted from the IPCC recommendations.
We introduce the following inputs and sets:

J = the set of demand nodes indexed by j
I = the set of candidate facility locations at the mid layer, indexed by i
K = the set of candidate facility locations at the high layer, indexed by k
h_j = demand at customer location $j \in J$
f_i = fixed cost of locating a mid layer facility at candidate site $i \in I$
g_k = fixed cost of locating a high layer facility at candidate site $k \in K$
c_{ij} = is the unit cost of supplying demand $j \in J$ from a mid layer facility located in $i \in I$
l_{ik} = is the unit cost of supplying demand $i \in I$ from a high layer facility located in $k \in K$
M = cardinality of J
α_i = GHG emissions factor of a facility located at candidate site $i \in I \, U \, K$, in tons of CO_2e per unit demand
β_{ij} = GHG emissions factor per unit distance and per unit demand between candidate facility site $i \in I$ and customer location $j \in J$, in tons of CO_2e per km and unit demand. We also use this factor for facilities located at a higher layer ($k \in K$) that are serving mid layer facilities ($i \in I$).
We consider the following decision variables:

y_i = 1 if we locate a facility at candidate site $i \in I$, (y_i = 0 otherwise)
z_k = 1 if we locate a high layer facility at candidate site $k \in K$, (z_k = 0 otherwise)
x_{ij} = 1 if the demand of $j \in J$ is serviced by a facility located at $i \in I$ (x_{ij} = 0 otherwise)
w_{ik}=1 if the demand of $i \in I$ is serviced by a facility located at $k \in K$, (w_{ik}=0 otherwise)

The general supply chain design problem with sustainable constraints ((*GUSSCP*) is defined by:

$$v(GUSSCP) =$$

$$Min \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} h_j c_{ij} x_{ij} + \sum_{k \in K} g_k z_k + \sum_{k \in K} \sum_{i \in I} h_i l_{ik} w_{ik} \quad (1)$$

Subject to

$$\sum_{i \in I} x_{ij} = 1 \qquad \forall j \in J \quad (2)$$

$$\sum_{j \in J} x_{ij} \leq M y_i \qquad \forall i \in I \quad (3)$$

$$\sum_{k \in K} w_{ik} \geq y_i \qquad \forall\, i \in I \qquad (4)$$

$$\sum_{i \in I} w_{ik} \leq z_k \qquad \forall\, k \in K \qquad (5)$$

$$\sum_{i \in I}\sum_{j \in J} \alpha_i \quad h_j x_{ij} + \sum_{k \in K}\sum_{i \in I} \alpha_k \quad h_i x_{ik} + \sum_{i \in I}\sum_{j \in J} \beta_{ij} h_j c_{ij} x_{ij} + \sum_{k \in K}\sum_{i \in I} \beta_{ik} h_i l_{ik} w_{ik} \leq GHG \qquad (6)$$

$$x_{ij}, w_{ik}, y_i, z_k \in \{0,1\} \qquad \forall\, i \in I, \forall\, j \in J, \forall\, k \in K \qquad (7)$$

The objective function (1) minimizes the sum of the installation facility costs and the demand-weighted supplying costs. Constraints (2) warranty that all demand is met. Constraints (3) warranty that a demand node j must be allocated to a facility $i \in I$ already opened. Constraints (4) ensure that each open mid layer facility must be allocated to one high layer facility. Constraints (5) warranty that the demand of a mid layer facility must be serviced by a high layer facility. Constraint (6) limits the total Greenhouse Gas (CO_2e) emissions to *GHG*. Constraints (7) are standard binary constraints. Regarding constraints (3), when they are replaced by constraints:

$$x_{ij} \leq y_i \qquad \forall\, i \in I, \forall\, j \in J \qquad (3a)$$

we got a stronger formulation for the problem, as it was also discussed in a related facility location problem [15]. This problem is NP-hard as generalization of well-known location problems, and therefore cannot be solved in polynomial time.

4 Genetic algorithm implementation

Genetic Algorithms (GAs) are a type of evolutionary algorithms (EVA) used to solve a number of combinatorial optimization problems. See for further details the papers [16,17]. According to [18], an EVA is composed by five basic components: (a) a genetic representation of solutions to a problem; (b) a way to create an initial population of solutions; (c) an evaluation function; (d) genetic operators that alter the genetic composition of children during reproduction and (e) values for the parameters. In this section we briefly describe these components and the GA implementation for solving the *GUSSCP* problem.

In our implementation, each solution (individual) to the problem is coded as a chromosome such that each gene corresponds to a facility location decision variable at both layers, taking value 1 if a facility is open and zero otherwise. We use one string to represent facilities located at both layers. The initial population of 100 individuals is generated randomly. Then we recombine this initial population and generate randomly two sets of 100 individual each. Each gene of the chromosome is generated by a 0-1 uniform probability distribution. For each chromosome we apply a solution procedure. This procedure consists in getting a solution to *GUSSCP* disregarding sustainable constraints as follows: Given a chromosome, for every customer $j \in J$ we assigned its nearest opened mid layer facility. Then for each opened mid layer facility we assigned its nearest opened high layer facility. Regarding the GHG emissions, that mechanism of generating chromosome could generate unfeasible solutions for *GUSSCP*, but our strategy was to explore the behavior of the algorithm based on an initial population composed of a number of unfeasible solutions. For our test problems, the GA implemented in this way rapidly generated a large number of unfeasible solution and we obtained poorer solutions than the next approach we will discuss it. In the second approach, after the crossover and mutation operations, we introduce a greedy-random procedure to generate to every iteration of the GA at least 50% of feasible solutions. Then our new population in every iteration has at least 50% of feasible solutions. The procedure is as follow: we generate a random individual consisting of a number of facilities opened/closed. Then based on the transportation costs we apply the solution procedure, i.e., we allocate the nearest customer to each opened mid layer facility and then for each mid layer facility we find the nearest opened high layer facility. We repeat the procedure till we get 50% of feasible solutions for the new population. We replace unfeasible solutions for the new individuals obtained through this procedure. Finally the best 100 individuals will be part of the new initial population to start the main iteration of GA. As we can see later in this paper, computational results are very good.

The fitness of a chromosome is calculated using the objective function (1). To compute the first term (installation costs) of (1) is straight forward from the chromosome. To compute the second term (transportation costs) of (1), we use the solution procedure described above: for each customer we find its nearest opened facility (minimal transportation cost) and the we do the same for each opened mid layer facility. Then we sum up both parts (installation and transportation costs) to get the objective function value for each individual of the population.

We use the standard genetic operators. The crossover generates two new individual (chromosome) exchanging the genetic material of two (parental) individuals expecting that "good" solutions can generate "better" ones. We selected these individuals randomly from a two sets of individuals, each set composed of 100 individuals as described earlier. We do not limit the number of new chromosomes generated by crossover. In this work crossover probability (*cross_p*) is set to 0.7 (70%) and we perform one-point crossover. The crossover procedure is simple, we generate a random value, if the *cross_p* value is greater than the random value then we pick one individual from each set. We generate another random value between one and the number of potential facility sites, i.e, a cut point dividing each individual (parent) into two segments. The first child is created by combining the first segment from the first parent and the second segment from the second parent. The second child is created from the first segment of the second parent and the second segment of

the first parent. The mutation operator changes the value of a chromosome with some small probability. In our case, we get this probability to 0.1 (10%). The gene in the chromosome is selected randomly and we switch its value (0-1). We do not limit the number of new chromosomes generated by mutation. The selection operator is based on elitist selection, favoring individuals of better fitness value to reproduce more often than the worse ones when generating the new population. In every iteration the whole population (200 individuals) is ranked in a non-decreasing order of the objective function value. As we described earlier, feasibility of constraints (5) is verified. In case there is lesser than 50% of feasible solutions in the population, a greedy-random procedure was implemented to generate new (feasible) individuals. The best (100) solutions passed to the next iteration
In our case the total size of population is 200 individuals, and 100 of new individuals are generated each iteration. We set the total number of iteration to six.

5 Computational results

Avoid The GA solution method for this problem was coded and implemented by Scilab software. According the sustainable supply chain literature discussed in previous section, for testing our GA implementation we generated 11 small size instances of *GUSSCP*. These instances correspond to test networks up to 26 potential sites and up to 50 demand nodes taken from the ORLIB [19]. As we do not know in advance how well is going to perform the GA, in order to validate our GA solutions we used GAMS on integer linear programming model described in section 3. Every test problem was running 5 times and we present an average value in Table 1. The optimal objective values were obtained by GAMS. As we can see in Table 1, besides the few number of iteration (6) used in our GA, the GAP obtained is quite small. Both methods (GA and GAMS) quickly converge on mentioned *GUSCPS* instances and their running times are not reported. The alpha (α) and beta (β) parameters were set to one and two respectively. This was done to analyze the behavior of the algorithm and also to check how the solution change when you penalty the transportation GHG emissions. Total GHG emissions were limited to values between 3,200.00 and 10,000.00 thousands.

We notice that, when you reduce the total amount of GHG emissions permitted, and the number of facilities remain free, the number of facilities to open increase, also increasing the cost of the solution but reducing the amount of GHG emitted by the transportation component.

6 Conclusions

In this paper, we introduced a novel kind of sustainable supply chain network design problem with a GHG emission constraint. The problem addressed the design of supply network arising mainly in the public sector, where we need to satisfy the demand for services like education and health care locating a number of facilities. We limit the GHG emissions generated by the facilities and the transportation involved in servicing the customers. The problem was formulated as a mixed integer 0-1 linear programming problem (MIP) and solved using a genetic algorithm coded in Scilab. We conducted an experimental study on instances of small sizes taken from the ORLIB. In order to validate our GA solutions we used GAMS to obtain optimal objective values on the MIP. The genetic algorithm performs very good considering we set a few number of iterations. We observed that when you reduce the total amount of GHG emissions permitted, and the number of facilities remain free, the number of facilities to open increase, also increasing the cost of the solution but reducing the amount of GHG emitted by the transportation component.

Table 1. Computational Results. First column indicates number of problem instance; Fixed Costs in millions; Total GHG emissions in millions ; z* is the optimal solution; z(GA) is the solution value provided by GA

# Prob	Fixed Costs	Total GHG	z*	z(GA)	GAP (%)
1	25.0	10.0	1,746,347	1,775,425	1.7
2	17.5	10.0	1,727,848	1,731,842	0.2
3	12.5	10.0	1,700,236	1,700,841	<0.1
4	25.0	5.0	1,746,348	1,775,425	1.7
5	25.0	3.2	1,953,224	1,953,224	0.0
6	17.5	3.2	1,840,724	1,840,724	0.0
7	12.5	3.2	1,765,724	1,765,724	0.0
8	7.5	3.2	1,690,724	1,690,724	0.0
9	7.5	3.3	1,663,018	1,684,578	1.3
10	12.5	3.3	1,700.236	1,765,723	3.9
11	17.5	3.3	1,730,236	1,773,011	2,5

Acknowledgments

This research was partially supported by the Fundação de Amparo á Pesquisa do Estado de Minas Gerais – FAPEMIG e CNPq - Brazil.

References

[1] D. Simchi-Levi, P. Kaminsky, and E. Simchi-Levi, *Designing & Managing the Supply Chain*. 3rd ed. Boston, MA, USA: McGraw-Hill/Irwin, 2007.

[2] S. Seuring, and M. Muller, "From a literature review to a conceptual framework for sustainable supply chain management", *Journal of Cleaner Production*, vol. 16, pp. 1699-1710, 2008.

[3] T. Y. Choi, and Z. Wu, "Taking the leap from dyads to triads: Buyer–supplier relationships in supply networks",

Journal of Purchasing and Supply Management, vol. 15, no. 4, pp. 263-266, 2009.

[4] S. Srivastava, "Green Supply-Chain Management: A State-of-the-Art Literature Review", *International Journal of Management Reviews*, vol. 9, no. 1, pp. 53-80, 2007.

[5] A. Hugo, and E. Pistikopoulos, "Environmentally conscious process planning under uncertainty", in *Sixth international conference on foundations of computer-aided process design*. C.A. Floudas, and R. Agrawal, Eds. Princeton, CACHE Corporation, 2004.

[6] A. Hugo, and E. Pistikopoulos, "Environmentally conscious long-range planning and design of supply chain networks", *Journal of Cleaner Production: Recent advances in industrial process optimization*, vol. 13, no. 15, pp. 1471-1491, 2005.

[7] A. Ramudhin, A. Chaabane, M. Kharoune, and M. Paquet, "Carbon market sensitive green supply chain network design", in *Proceedings IEEE International Conference on IEEE*, Singapore, 2008, pp. 1093-1097.

[8] A. Chaabane, A. Ramudhin, and M. Paquet, "Optimization and evaluation of sustainable supply chains", in *Proceedings of the 8th. International Conference of Modelling and Simulation* – MOSIM 2010, Tunisia, 2010.

[9] A. Diabat, and D. Simchi-Levi, "A carbon-capped supply chain network problem", in *Proceedings of IEEE International Conference on Industrial Engineering and Engineering Management 2009*. IEEE, Piscataway, N.J., 2010, pp. 523-527.

[10] M. S. Daskin, L. V. Snyder, and R. T. Berger, "Facility Location in Supply Chain Design", in *Logistics Systems: Design and Optimization* , A. Langevin and D. Riopel, Eds. Kluwer, 2005, pp. 39-65.

[11] F. Altiparmak, M. Gen, L. Lin, and T. Paksoy, "A genetic algorithm approach for multiobjective optimization of supply chain networks", *Computers & Industrial Engineering*, vol. 51, pp. 197–216, 2006.

[12] R. Z. Farahani, and M. Hekmatfar, Eds., *Facility location: concepts, models, algorithms and case studies*. Berlin, Germany, Springer-Verlag, 2009.

[13] W. Lai-Jun, S. Xiao-Ling, and S. Zhongke, "New evolutionary Algorithm Applying to a Type of Facility Location Problem", *Information Technology Journal*, vol. 8, no. 4, pp. 605-609, 2009.

[14] S. Bastani, and N. Kazemzadeh, "Hierarchical Location Problem", in *Facility location: concepts, models, algorithms and case studies*. *Contributions to Management Science*, R.Z. Farahani and M. Hekmatfar, Eds. Springer, 2009, pp. 219-241.

[15] G. Cornuejols, M. L. Fisher, and G. L. Nemhauser, "Location of bank accounts to optimize float: An analytic study of exact and approximate algorithms", *Management Science*, vol. 23, pp. 789–810, 1977.

[16] K. E. De Jong, and W. M. Spears, "Using Genetic Algorithms to Solve NP-Complete Problems", in *Proceedings of the Third International Conference on Genetic Algorithms*, San Mateo, California, Morgan Kaufmann, 1989, pp, 124-132.

[17] D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, 1989.

[18] I. H. Osman, and J. P. Kelly, *MetaHeuristics: Theory & Applications*. Kluwer Academic Publishers, 1996.

[19] J. E. Beasley, "Obtaining test problems via Internet", *Journal of Global Optimization*, vol. 8, pp. 429-433, 1996.

Comparing Genetic Algorithms and Simulated Annealing for Solving the Pickup and Delivery Problem with Time Windows

M. Hosny[1]
[1]College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia

Abstract— *Solving the Vehicle Routing Problem (VRP) and its related variants is significant for optimizing logistic planning. One important variant of the VRP is the Pickup and Delivery Problem with Time Windows (PDPTW), where it is assumed that a client may request one of two service types, a pickup or a delivery. It is required to find a set of minimum cost routes for a fleet of vehicles, while observing a number of predefined constraints. In this research, we try to handle the difficult constraints using a new solution representation and simple neighborhood moves that will maintain the feasibility of solutions throughout the search. Our solution method is tried within two meta-heuristic approaches, a Genetic Algorithm and a Simulated Annealing. Based on the performance of the two algorithms on a number of benchmark problems, we draw conclusions about which among the two algorithms seems more appropriate for solving the problem.*

Keywords: Pickup and Delivery, Vehicle Routing, Heuristics, Meta-heuristics, Combinatorial Optimization

1. Introduction

The Pickup and Delivery Problem with Time Windows (PDPTW) is one important variant of the Vehicle Routing Problem with Time Windows (VRPTW). Unlike the VRPTW, where all requests are of the same type, the PDPTW assumes that requests are paired, and for each pickup request there is a corresponding delivery request. It is also assumed here that there is a number of identical vehicles, operating from a central depot, that should serve the set of requests. It is required to find a set of minimum cost routes for the vehicles, such that the following constraints are satisfied: (1) the precedence constraint, which requires that each pickup location must be visited before its corresponding delivery location (2) the capacity constraint, in which the total load carried by each vehicle at all times should not exceed its capacity and (3) the Time Window (TW) constraint in which it is necessary to visit each location within a predetermined time interval, such that if the vehicle arrives at a location before its due service time, it must wait until the beginning of the specified period. A formal problem definition can be found in [1]. Applications of the PDPTW are frequently encountered in every day transportation and logistic services, and the problem is likely to assume even greater prominence in the future, due to the increase in e-commerce and Internet shopping.

Similar to the VRPTW, the PDPTW is *NP-hard* [2]. However, the constraints imposed in the problem make finding a suitable solution algorithm a hard challenge for researches. Since exact algorithm are often very slow when applied to practical problem sizes, heuristic and meta-heuristic approaches are usually the preferred solution methods. The solution approach mainly proceeds in two stages: (1) solution construction, in which one or more initial problem solutions is created and (2) solution improvement, in which the initial solution(s) is modified using a heuristic or a meta-heuristic algorithm. In addition, there are generally two aspects to consider while solving this problem: the grouping part, which involves assigning requests to vehicles, and the routing part, in which each individual vehicle route must be planned such that all constraints are adhered with.

In our research, we mainly focus on how to represent and construct a problem solution in a simple way without violating all problem constraints. We also focus on designing intelligent neighborhood moves that preserve the feasibility of the solution throughout the search. Unlike most previous research in this area, our technique does not rely on a parallel repair algorithm, to fix solution infeasibility. We tried our suggested approach within two meta-heuristics: Genetic Algorithms (GAs) and Simulated Annealing (SA) and compared their performance on some benchmark instances. We also compared the results to best known results from the literature. Based on the results obtained, we try to highlight the promising aspects of our approach and also demonstrate where further improvement can be achieved.

The rest of this paper is organized as follows: Section 2 summarizes some related work. Section 3 explains the GA approach. Section 4 explains the SA approach. Section 5 shows the experimental results, and finally Sect. 6 concludes with some suggested future directions.

2. Related Work

The authors in [1] present a Tabu-embedded Simulated Annealing with K-restarts, i.e, the algorithm stops when the number of iterations without improvement reaches a predefined value K. To prevent cycling, their SA procedure records the accepted solutions in a tabu list. The authors created a number of test cases, which have since then been

used as benchmark data to test solution methods. In [3], the authors present a two-stage algorithm to deal with the PDPTW. The first stage uses a simple SA approach, whose focus is to minimize the number of vehicles used in the solution. The neighborhood move adopted in their SA is a simple pickup and delivery pair relocation operator. The second stage uses a Large Neighborhood Search (LNS) strategy, to minimize the total travel cost of the entire solution. The LNS is actually a sequence of local searches, where a large number of requests (30% - 40%) are first removed then re-inserted into the solution.

An interesting solution methodology is presented in [2], where the basic solution methodology adopts the LNS technique, used in [3]. Unlike previous researchers, though, the authors apply a number of different heuristics for both the removal and the insertion methods of the selected requests. They decide among these heuristics adaptively during the search. Accordingly, the authors call their algorithm an Adaptive Large Neighborhood Search (ALNS). During the search, a newly generated solution replaces the current solution using an SA acceptance criterion. Similar to [3], they first apply a preliminary stage to minimize the number of vehicles used in the solution before applying the basic ALNS.

The authors in [4] introduce an attempt to handle the PDPTW using an evolutionary algorithm, for both the grouping and the routing aspects of the problem. The solution representation is a list of vehicles routes, where each route consists of a sequence of pickup and delivery requests. Two crossover operators are tried, one exchanges fragments of routes between parents, while the other exchanges complete routes. Infeasible solutions that may be created following crossover are repaired by removing repeated requests or adding requests that are not served. If this is not possible, the offspring is discarded. Two mutation operators are presented, the first tries to reduce the number of vehicles in a solution by selecting a route and moving all its requests to other routes in the same solution. The second mutation tries to improve a route by rearranging its requests.

The work in [5] presents a grouping genetic algorithm (GGA) to the PDPTW. In a GGA, the genetic representation is based on a set of genes, where each gene represents a group of objects rather than a single object. For the PDPTW, it is assumed that each gene represents a group of requests that are assigned to one vehicle. Thus an individual solution only covers the grouping aspect of the problem. The routing aspect, on the other hand, is handled by an independent data structure associated with each gene. In the crossover, clusters (vehicles with their assigned requests) are removed from one parent and inserted into the other parent. This is then followed by a chromosome cleanup, to remove duplicate vehicles and the repeated assignment of requests, in addition to re-assigning requests that are no longer assigned. The mutation operator removes a cluster from a chromosome, and re-assigns its requests to other clusters. The embedded insertion heuristic applied in several stages of this algorithm is based on inserting a request in the best feasible and minimum cost position, among all possible insertion positions in the chromosome.

The above brief literature summary clearly shows that the PDPTW is in fact a complicated problem, and developing an appropriate solution technique is often a hard challenge for researchers. In both the construction and the improvement phases of the problem, the researcher is faced with many decisions that should be made. These decisions include, among others, the components of the objective function, the permissibility of infeasible solutions, how to adhere with the constraints of the problem, how to generate new solutions, and what acceptance criteria should be applied to replace the current solution, during the improvement phase. It is thus often difficult to reach solutions that do not violate any problem constraint, and at the same time be satisfactory in terms of reducing total cost, in a reasonable processing time. Moreover, many published solution techniques usually tend to be rather complicated, and far from being intuitive or direct. In the following sections, we describe in detail our suggested approach to overcoming some of these difficulties.

3. The GA Approach

Research using GAs for solving the PDPTW is generally scarce, and the results reported by most GA techniques attempted are often disappointing one way or another. When trying to solve this problem using a GA, it is hard to tackle both the grouping and the routing aspects simultaneously. Moreover, a major issue is finding a suitable genetic encoding and designing intelligent genetic operators that are capable of handling all the difficult problem constraints [5]. Infeasible solutions are usually handled using a repair method to fix the infeasibility during the search, which will inevitably increase the processing time and complicate the algorithm. Most previous GA research, for example [5], tried to solve the problems encountered in the GA encoding and operators by allowing the GA to handle only the grouping aspect. The routing aspect, on the other hand, was handled by an independent routing algorithm that is hidden from the GA and is called when a chromosome is decoded. Other attempts to use a GA for both the grouping and the routing aspects, for example [4], generally produced discouraging results.

Our GA approach to solving the PDPTW was first presented in [6]. For convenience, we repeat in this paper the important details of this GA approach. The most distinguishing feature of our GA is that it tries to face the challenge of handling both the routing and the grouping aspects of the problem simultaneously. Unlike many popular approaches, in which the GA is only aware of how requests are clustered, but is not aware of how they are routed, our chromosome representation more naturally accommodates each group of

requests together with their suggested routes. The algorithm has an embedded simple routing heuristic that allows individual routes to dynamically change, within the chromosome itself, during the search . Also, we developed new simple genetic operators. Using problem-specific knowledge, such as the quality of the generated routes, these operators try to create good quality feasible solutions throughout the search. In addition, since no parallel repair method is needed to fix the infeasibility of solutions, the overall algorithm is simple and elegant, a feature often missing from most up-to-date solution algorithms.

3.1 The Solution Representation and the Objective Function

Following our approach in [7], for solving the Single Vehicle PDPTW (SV-PDPTW), we adopt a simple representation for each individual route. A route is simply a list of visited requests[1]. However, when we assign requests to each route, both the pickup and its delivery location are given the same code. A simple parser is then used to traverse the route and retrieve the information of the pickup location if this is the first encounter of the code, and retrieve the delivery information if this is the second occurrence. This way, the precedence constraint, between the pickup and the delivery, will always be satisfied, and will not be disturbed by any neighborhood move (e.g. a node swap), attempted to improve the route. An example of a route with 4 requests following this representation is: (**2 1** *1* **3 4** *2 3 4*), where pickups are shown in boldface and deliveries in italics.

In our GA, the *chromosome* represents a complete problem solution. It is simply a collection of individual routes. Thus, each gene is a complete vehicle route. Thus, our representation is not actually an encoding in the usual GA sense, rather it is just a problem solution upon which the genetic operators are directly applied. Following is an example of a chromosome consisting of 4 vehicles and 12 requests (each having a pickup and a delivery location), while node 0 represents the depot.

Chromosome:

$v1:$	0	1	2	3	3	2	1	4	4	0
$v2:$	0	5	5	6	6	7	7	0		
$v3:$	0	8	9	8	9	0				
$v4:$	0	10	11	12	11	10	12	0		

Similar to most solution methods in the literature, for example [1], our *objective function* tries first to minimize the number of vehicles used in the solution followed by both the total distance traveled and the total route duration. We used the following objective function of a solution S to achieve this goal:

$$O(S) = N^2 \times TotDist(S) \times TotDur(S) \qquad (1)$$

[1]Hereafter we use the term *request* to refer to a pickup and delivery pair of related locations.

where N is the number of vehicles, $TotDist(S)$ is the total distance traveled by all vehicles, and $TotDur(S)$ is the total schedule duration, which includes the total travel time, the waiting time of the vehicles, and the service time at each location.

3.2 The Initial Population

The initial population is generated using a sequential construction algorithm to create each solution (an individual in the population). The algorithm utilizes a simple routing heuristic to achieve fast and feasible vehicle routes. This sequential construction algorithm is in fact used in many parts of our GA during the evolutionary process, as will be explained later while addressing the GA operators. In what follows, we will first explain the routing algorithm that is concerned with individual vehicle routes, and then explain the overall sequential construction process that is used to create a complete problem solution.

The Routing Algorithm

Our routing algorithm, described in Algorithm 1, was first introduced in [7], for the SV-PDPTW. Unlike most other routing (insertion) algorithms in the literature, for example [8], our routing heuristic does not try to calculate the insertion cost at each and every possible insertion position, and select the one with the least cost. Rather, it only tries to improve the current route by using a simple Hill-Climbing (HC) heuristic. It is thus very fast and much simpler than the 'classical' insertion heuristics. The neighborhood move used in the HC algorithm is a regular swap of two locations. However, in order to satisfy the hard TW constraint, our neighborhood move only swaps locations if the latter location has a TW deadline that precedes the earlier location.

Algorithm 1 The Routing Algorithm

```
1: Given a route r
2: repeat
3:    for (Each possible pair of locations in r) do
4:       if (The latter location is more urgent in its upper time
         window bound) then
5:          Swap the current 2 locations in r to get a new route
            r'
6:          Δ ← cost(r') − cost(r)
7:          if (Δ < 0) then
8:             Replace r with r'
9: until (Done){Stop when no improvement has been achieved in
   the previous pass}
```

The cost function, in Step 6 of the HC algorithm, is used to evaluate the quality of each route. This cost function tries to minimize the total route duration as well as the degree of infeasibility in capacity and TW constraints. The cost function of a route r is described by the following equation:

$$F(r) = w_1 \times D(r) + w_2 \times TWV(r) + w_3 \times CV(r) \quad (2)$$

where $D(r)$ is the total route duration, including the waiting time and the service time at each location. $TWV(r)$ is the total number of time window violations in the route, and $CV(r)$ is the total number of capacity violations. The constants w_1, w_2, and w_3 are weights in the range $[0, 1]$, and $w_1 + w_2 + w_3 = 1.0$. The choice of appropriate weights depends on the importance of each term in the objective function. We found that in order to get feasible solutions, the largest penalty should be imposed on the TW violations.

The Sequential Construction Algorithm

We used a simple sequential construction algorithm (*SEQ*), that we first suggested in [9], to construct a solution (an individual in the population) for our GA. To construct a solution, requests are first placed in a relocation pool in a random order. The *SEQ* algorithm then starts with one empty route, and a request from the pool (pickup and delivery pair) is initially inserted at the end of the current route. The routing algorithm (Algorithm 1) is then called to improve the current route and return a new route. Afterwards, the *SEQ* algorithm will check the new route for feasibility. If the new route is feasible, the insertion is accepted and the next request in order will be tried for insertion. On the other hand, if the new 'improved' route is still not feasible, the newly inserted couple will be removed from the route and remain in the relocation pool to wait for a later insertion attempt in a new route. This sequential construction process is shown in Algorithm 2.

Algorithm 2 The Sequential Construction (*SEQ*) Algorithm

```
1: Let M ← 0 {M is the number of vehicles used in the current solution}
2: repeat
3:    Initialize an empty route r
4:    M = M + 1
5:    for (All unassigned requests) do
6:       Get the next unassigned request i
7:       Insert the request i at the end of the current route r
8:       Call the HC routing heuristic (Algorithm 1) to improve r
9:       if (r is a feasible route) then
10:         Mark i as inserted
11:      else
12:         Remove i from r
13: until (All requests have been inserted)
```

3.3 The Genetic Operators

Mutation

The mutation operator, which we will call the *Vehicle Merge Mutation (VMM)*, simply tries to merge two vehicles selected at random. The idea is to try to reduce the number of vehicles by distributing the requests among already existing vehicles, or possibly combining the two vehicles into one. For example, assume that the current solution contains the following vehicles:

$v1:$	0	1	2	3	3	2	1	4	4	0
$v2:$	0	5	5	6	6	7	7	0		
$v3:$	0	8	9	8	9	0				
$v4:$	0	10	11	12	11	10	12	0		

Now Assume that vehicles $v2$ and $v3$ were selected for merging. The requests belonging to them will now be placed in a relocation pool in a random order.

Relocation Pool: 8 7 6 5 9

The remaining requests in the solution, i.e., vehicles $v1$ and $v4$ will be copied to the new solution to form a partial solution.

$v1:$	0	1	2	3	3	2	1	4	4	0
$v2:$	0	10	11	12	11	10	12	0		

The requests in the relocation pool are then re-inserted into the partial solution using the SEQ construction, i.e., Algorithm 2, and the final solution is constructed.

$v1:$	0	1	2	3	1	3	**7**	2	4	**7**	4	0
$v2:$	0	10	11	12	**9**	11	10	**9**	12	**6**	**6**	0
$v3:$	0	**5**	**8**	**5**	**8**	0						

Crossover

Two crossover operators have been used in our research. The first crossover operator, which we will call the *Vehicle Merge Crossover (VMX)*, is similar to the mutation operator described previously. However, instead of merging two vehicles from the same solution, the *VMX* tries to merge two vehicles selected at random, one from each parent solution. The second crossover operator, which will call the *Vehicle Copy Crossover (VCX)*, tries to copy complete routes from the parent to the child. The number of routes to be copied is a random number between $1/4$ to $1/2$ the number of routes in the first parent. To select routes for inheritance, the *VCX* tries to select the 'good' routes. It is generally desirable to copy routes that serve a large number of requests, since our main objective is to reduce the number of vehicles. Accordingly, the *VCX* first ranks routes based to the number of nodes served in each route. The larger the number of nodes served the higher the rank of the route. Routes with the same number of nodes are ranked according to the total distance traveled, in which case routes with a shorter distance are more favorable than the longer ones. To illustrate the *VCX*, consider the following example:

Parent1:

$v1:$	0	1	2	3	3	2	1	4	4	0
$v2:$	0	5	5	6	6	7	7	0		
$v3:$	0	8	9	8	9	0				
$v4:$	0	10	11	12	11	10	12	0		

Parent2:

$v1:$	0	**5**	**6**	**6**	1	1	**5**	0
$v2:$	0	2	**7**	2	3	3	**7**	0
$v3:$	0	4	4	12	12	0		
$v4:$	0	**8**	**9**	**8**	**9**	0		
$v5:$	0	10	10	11	11	0		

Now assume that vehicles $v1$ and $v4$ were selected from

Parent1, depending on the ranking criterion described above. These two vehicles will now be copied to the first child.

Child1:

$v1:$ 0 1 2 3 3 2 1 4 4 0

$v2:$ 0 10 11 12 11 10 12 0

The remaining requests that have not been included in Child1 (shown in boldface in Parent2) will be copied in the same order of their appearance in Parent2, and placed in a relocation pool.

Relocation Pool: 5 6 7 8 9

The requests in the relocation pool are then sent to the solution construction algorithm and used to form a set of new routes. These new routes will afterwards be appended to the routes already existing in Child1, which were inherited from Parent1.

Child1:

$v1:$ 0 1 2 3 3 2 1 4 4 0

$v2:$ 0 10 11 12 11 10 12 0

$v3:$ **0 5 6 8 9 8 9 5 6 0**

$v4:$ **0 7 7 0**

Child2 is created similarly by reversing the roles of parents.

4. The SA Approach

We will try in this part of our research to adopt some operators used in the GA approach for the PDPTW within an SA approach. The solution representation used in the SA is similar to the chromosome representation used in the GA (explained in Section 3.1). The same objective function (1) is also used here. The initial solution from which the SA will progress is created by generating a number of random solutions, in a manner similar to the creation of an initial genetic population (as explained in Section 3.2), and the best solution among them, in terms of the objective function value, is selected.

To allow for an adaptive calculation of the SA parameters for each problem instance individually, we used the approach proposed by [10]. Thus, the annealing parameters are calculated based on the average value of $\Delta cost$, where $\Delta cost$ is the difference in the objective function value between some randomly generated solutions for the current problem instance. Also, a crucial part of any SA algorithm is the neighborhood move that will be used to generate a new solution. We experimented with different neighborhood moves from the ones tried in our GA approach and other popular neighborhood moves from the literature. Two neighborhood moves were found to be the most appropriate for the SA approach:

The Large Neighborhood Search (LNS) Move: The LNS move is inspired from the Adaptive Large Neighborhood Search (ALNS) approach of [2]. The idea is to remove and then re-insert a large number of requests in each application of the move. The authors in [2] recommend that 30% to 40% of the total number of nodes is removed in each iteration. In our LNS move, we slightly increased the allowed range for the number of removed requests, in order to explore a wider area of the search space. The number of removed requests in our LNS move ranges between 20% to 50% of the total number of requests. In addition, we adopted here the "Worst Removal" variant from the three types of removal heuristics applied in [2]. In our approach, the requests removed are those that are estimated to cause a large increase in the cost of their respective routes. Thus, to determine the cost of each request, the total travel distance of the route to which the request belongs is calculated, with and without the request under consideration, and the difference in the route distance is used as a measure of the request cost. Requests having the highest costs are selected for removal from the current solution. After this, the removed requests are placed in a relocation pool in a random order, before they are inserted back in the solution using the SEQ construction algorithm, hoping to find better insertion positions for them in the new solution.

The Vehicle Merge (VM) Move: This move is identical to the Vehicle Merge Mutation (VMM), explained in Section 3.3.

Our SA approach to the PDPTW operates in two stages. In the first stage, the LNS move is used to generate a new solution, while in the second stage the VM move is used instead. Changing the SA move in this manner allows the search process to slightly perturb the current solution, before trying to re-optimize it, which may help in escaping local optima. The second SA stage starts from the final solution obtained in the previous stage and from the final temperature reached by the end of the previous stage. During each stage, the best so far solution is saved and each SA stage terminates when no improvement is realized in the best solution for a consecutive number of iterations. Also, during each stage, the current temperature value is reduced in each iteration of the SA algorithm. For a more extensive searching, the two SA stages are repeated several times. The repetition only stops when the best obtained solution reaches a stage of stagnation and does not improve for a number of consecutive attempts of applying the two stages. We also found during our computational experimentation that there seems to be no significance to the order of application of the two moves. Thus, as long as the two stages are repeated, the SA may be started from either move. The overall 2-stage SA approach is shown in Algorithm 3.

5. Experimental Results

We used the 56 (100-customers) benchmark instances, created by Li & Lim in [1]. There are 6 different categories of problem instances: *LC1*, *LC2*, *LR1*, *LR2*, *LRC1* and *LRC2*. Problems in the *LC* category have clustered customers, problems in the *LR* category have randomly

Algorithm 3 The 2-Stage SA Algorithm

1: Find an initial solution ($InitSol$) and calculate the annealing parameters
2: $BestSol \leftarrow InitSol$ {Initialize the best so far solution}
3: Initialize $MaxAttempts$ to a small number {We used 10}
4: $NoImprovement \leftarrow 0$
5: **repeat**
6: $OldCost \leftarrow Objective(BestSol)$ {Calculate the cost of the solution using Equation (1)}
7: $BestSol \leftarrow SA_{LNS}(BestSol)$ {Perform SA on the current best solution, using the LNS move, and return the best found solution}
8: $BestSol \leftarrow SA_{VM}(BestSol)$ {Perform SA on the current best solution, using the VM move, and return the best found solution}
9: $NewCost \leftarrow Objective(BestSol)$ {Calculate the cost of the new solution using Equation (1)}
10: **if** ($NewCost$ is not better than $OldCost$) **then**
11: $NoImprovement++$
12: **else**
13: $NoImprovement \leftarrow 0$
14: **until** ($NoImprovement$ reaches $MaxAttempts$)

distributed customers, and problems in the *LRC* category have both random and clustered customers. Also, problems identified with '1' have a tight TW width, while problems identified with '2' have a large TW width. The data together with the best-known results can be downloaded from: `http://www.top.sintef.no/vrp/benchmarks.html`. Each algorithm was run 10 times on each problem instance. Table 1 shows the best result, in the 10 runs, achieved by both the GA and the SA for each test case. The best result is the one having the minimum number of vehicles, and for the same number of vehicles, the one having the minimum total travel distance. The better obtained result between the two algorithms is highlighted in boldface. The last two columns of the table show the current best known results. The last two rows of the table show, respectively, the overall average results of the corresponding column, and the percent difference (gap) between our average and the average of the best known results, in terms of both the number of vehicles and the total distance.

Table 1 shows that SA achieved better results than the GA in 31 out of the 56 problem instances. In terms of the overall average results, SA achieved a slightly better average in the number of vehicles and a slightly worse average in the total distance traveled, which can also be seen from the relative difference to best known results, indicated in the last row of the table. In terms of the average processing time, though, the result was in favor of SA. The average processing time needed by the SA algorithm was 124.4 seconds, while the GA needed an average processing time of 226.2 seconds, i.e., the SA algorithm had an approximately 45% less average processing time than the GA.

In general the results in Table 1 indicate that both the SA and GA obtained comparable results when tested on

Table 1: GA & SA Best Results (100-Customers)

	GA		SA		Best Known	
Name	**Vehic**	**Dist**	**Vehic**	**Dist**	**Vehic**	**Dist**
LC101	**10**	**828.94**	**10**	**828.94**	10	828.94
LC102	11	978.48	**11**	**945.88**	10	828.94
LC103	10	1310.3	**10**	**1238.57**	9	1035.35
LC104	10	1203.1	**9**	**1328.29**	9	860.01
LC105	**10**	**828.94**	**10**	**828.94**	10	828.94
LC106	10	844.58	**10**	**828.94**	10	828.94
LC107	**10**	**828.94**	**10**	**828.94**	10	828.94
LC108	10	949.96	**10**	**828.04**	10	826.44
LC109	**10**	**827.82**	10	849.08	9	1000.6
LC201	**3**	**591.56**	**3**	**591.56**	3	591.56
LC202	5	1261.81	**4**	**1186.64**	3	591.56
LC203	5	1957.86	**5**	**1903.04**	3	585.56
LC204	5	1770.26	**4**	**2194.2**	3	590.6
LC205	**3**	**591.56**	**3**	**591.56**	3	588.88
LC206	4	681.35	**4**	**626.89**	3	588.49
LC207	3	766.62	**3**	**701.72**	3	588.29
LC208	**3**	**604.51**	3	604.7	3	588.32
LR101	**19**	**1667.68**	**19**	**1667.68**	19	1650.8
LR102	**17**	**1627.73**	17	1627.91	17	1487.57
LR103	14	1619.68	**14**	**1525.99**	13	1292.68
LR104	**11**	**1262.3**	11	1335.91	9	1013.39
LR105	**15**	**1433.79**	15	1450.98	14	1377.11
LR106	14	1564.06	**13**	**1458.71**	12	1252.62
LR107	12	1356.72	**12**	**1353.44**	10	1111.31
LR108	13	1380.93	**12**	**1353.05**	9	968.97
LR109	**13**	**1448.14**	13	1449.38	11	1208.96
LR110	13	1362.74	**12**	**1323.12**	10	1159.35
LR111	13	1431.07	**12**	**1299.28**	10	1108.9
LR112	12	1339.66	**11**	**1237.92**	9	1003.77
LR201	**4**	**1783.1**	4	1841.72	4	1253.23
LR202	5	2035.85	**4**	**2083.77**	3	1197.67
LR203	5	2135.47	**4**	**2312.46**	3	949.4
LR204	**4**	**2017.64**	4	2059.11	2	849.05
LR205	**4**	**1939.61**	4	2123.62	3	1054.02
LR206	**4**	**2128.15**	4	2134.15	3	931.63
LR207	**4**	**2191.28**	4	2360.2	2	903.06
LR208	4	2064.23	**4**	**2059.02**	2	734.85
LR209	5	1787.82	**4**	**2058.16**	3	930.59
LR210	5	1928.42	**4**	**2175.39**	3	964.22
LR211	**4**	**1782.99**	4	1886.82	2	911.52
LRC101	**16**	**1806.27**	**16**	**1806.27**	14	1708.8
LRC102	15	1840.05	**14**	**1776.15**	12	1558.07
LRC103	13	1599.7	**13**	**1558.38**	11	1258.74
LRC104	13	1535.82	**12**	**1464.11**	10	1128.4
LRC105	16	2009.51	**16**	**1862.36**	13	1637.62
LRC106	**13**	**1688.86**	14	1672.87	11	1424.73
LRC107	14	1637.27	**12**	**1452.42**	11	1230.15
LRC108	13	1535.48	**12**	**1379.59**	10	1147.43
LRC201	**5**	**2230.74**	5	2327.42	4	1406.94
LRC202	**5**	**2442.78**	5	2525.27	3	1374.27
LRC203	**5**	**2335.04**	5	2413.63	3	1089.07
LRC204	5	2049.8	**4**	**2323.05**	3	818.66
LRC205	6	2162.85	**5**	**2605.27**	4	1302.2
LRC206	**5**	**2181.3**	5	2400.12	3	1159.03
LRC207	5	2346.32	**5**	**2319.27**	3	1062.05
LRC208	**5**	**1983.58**	5	2195.22	3	852.76
AVG	**8.75**	**1562.52**	**8.43**	**1592.23**	**124.37**	**1036.68**
GAP%	**22%**	**51%**	**17%**	**54%**	-	-

the 100-customers instances for the PDPTW. This could be explained by realizing that both algorithms adopted similar neighborhood moves. The VM move used in SA is exactly the same as the VMM mutation used in the GA. In addition, the LNS move used in SA is also comparable to the crossover operators used in the GA. In the GA, the crossover operators generally try to remove requests belonging to some selected routes from parent solutions and insert them back into the child solution. This process is similar to what the LNS move performs in SA, when requests are removed and then re-inserted into the solution. Moreover, selecting 'good' routes to be transferred to the child in the VCX crossover is also analogous to removing 'bad' requests in the LNS move adopted in the SA. However, the fact that SA was much faster than the GA and obtained an overall better average results, in terms of the main objective (minimizing the number of vehicles), indicates that SA seems to be the more preferred approach to adopt for this problem. Implementation wise, SA is also easier to implement and may be less parameter dependent than the GA for solving the PDPTW.

The results of both algorithms, though, were inferior to the best known results, especially in terms of the total travel distance. This indicates that both algorithm could probably make use of an additional local search method, such as 2-Opt edge exchange, to improve the quality of the routes. Also, different neighborhood moves could be attempted in the route improvement heuristic (Algorithm 1), for example by taking some route characteristics into consideration when swapping locations. Despite this, both approaches are in fact simpler than many approaches attempted in the literature, and the representation and neighborhood moves suggested can be easily adapted to fit the different constraints of other vehicle routing problems.

6. Conclusions

In this paper we investigated both a GA and an SA for solving the PDPTW. Our GA approach tried to face the challenge of allowing the GA and its operators to be aware of and manipulate both the grouping and the routing aspects of the problem. Our operators overcome the difficult problem constraints, and avoid at the same time the need for a repair method to fix infeasible solutions, a technique that previous GAs and most other heuristic and meta-heuristic techniques have been relying on to maintain feasibility. The GA was compared with a 2-stage SA approach that uses two different neighborhood moves repeatedly. The neighborhood moves used within this SA approach are analogous to the genetic operators used previously in the GA, since they rely on removing and then re-inserting requests or merging selected vehicles.

Both the GA and the SA obtained comparable results when tested on published benchmark instances, which is mostly due to their reliance on similar neighborhood operators. However, the SA approach produced slightly better average result than the GA in terms of reducing the number of vehicles used. In addition, it was definitely much faster than the GA. This indicates that adopting SA is more recommended than adopting the GA for solving the PDPTW, since the overhead of maintaining a large population of solutions does not seem to be worthwhile, in terms of improving the final result obtained.

However, neither algorithm was capable of competing favorably with best known results. In general, it appears that our representation and neighborhood operators in both algorithms are doing a fair job in guiding the search towards better solutions. However, to cope with the difficulty of the problem and the different types of problem instances, both approaches still need further improvement. For this purpose, different neighborhood moves may be tried when improving individual vehicle routes. Moreover, a local search method, such as 2-Opt or 3-Opt, could be added to improve the quality of the routes. The operators and neighborhood moves suggested in this research, though, are portable and can be easily modified and utilized for solving different types of vehicle routing problems.

References

[1] H. Li and A. Lim, "A metaheuristic for the pickup and delivery problem with time windows," in *Proceedings of the 13th IEEE International Conference on Tools with Artificial Intelligence*, November 2001, pp. 160–167, dallas, TX, USA.

[2] S. Ropke and D. Pisinger, "An adaptive large neighbourhood search heuristic for the pickup and delivery problem with time windows," *Transportation Science*, vol. 40, no. 4, pp. 455–472, November 2006.

[3] R. Bent and P. Van Hentenryck, "A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows," *Computers & Operations Research*, vol. 33, no. 4, pp. 875–893, 2006.

[4] J. Créput, A. Koukam, J. Kozlak, and J. Lukasik, *Computational Science- ICCS 2004*. Springer, 2004, vol. 3038/2004, ch. An Evolutionary Approach to Pickup and Delivery Problem with Time Windows, pp. 1102–1108.

[5] G. Pankratz, "A grouping genetic algorithm for the pickup and delivery problem with time windows," *OR Spectrum*, vol. 27, pp. 21–24, 2005.

[6] M. I. Hosny and C. L. Mumford, "Investigating genetic algorithms for solving the multiple vehicle pickup and delivery problem with time windows," in *MIC2009, Metaheuristic International Conference*, July 2009.

[7] ——, "The single vehicle pickup and delivery problem with time windows: Intelligent operators for heuristic and metaheuristic algorithms," *Journal of Heuristics, Special Issue on Advances in Metaheuristics*, vol. 16, no. 3, pp. 417–439, June 2010.

[8] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Operations Research*, vol. 35, no. 2, pp. 254–265, 1987.

[9] M. I. Hosny and C. L. Mumford, "New solution construction heuristics for the multiple vehicle pickup and delivery problem with time windows," in *MIC2009, Metaheuristic International Conference*, July 2009.

[10] J. Dorband, C. L. Mumford, and P. Wang, "Developing an ace solution for two-dimensional strip packing," in *18th International Parallel and Distributed Processing Symposium Workshop on Massively Parallel Processing*, 2004, santa Fe, New Mexico.

SESSION

NATURAL LANGUAGE PROCESSING AND RELATED ISSUES

Chair(s)

TBA

PEN: Parallel English-Persian News Corpus

Mohammad Amin Farajian
Advanced Information and Communication Technology Research Center,
Sharif University of Technology, Tehran, Iran

Abstract - *Parallel corpora are the necessary resources in many multilingual natural language processing applications, including machine translation and cross-lingual information retrieval. Manual preparation of a large scale parallel corpus is a very time consuming and costly procedure. In this paper, the work towards building a sentence-level aligned English-Persian corpus in a semi-automated manner is presented. The design of the corpus, collection, and alignment process of the sentences is described. Two statistical similarity measures were used to find the similarities of sentence pairs. To verify the alignment process automatically, Google Translator was used. The corpus is based on news resources available online and consists of about 30,000 formal sentence pairs.*

Keywords: Parallel corpus, alignment, statistical machine translation.

1 Introduction

A corpus is a collection of pieces of language text in electronic form, selected according to external criteria to represent a language or language variety as a source of data for linguistic research [1]. A corpus may contain texts in a single language (monolingual corpus) or text data in multiple languages (multilingual corpus). Multilingual corpora that have been specially formatted for side-by-side comparison are called *aligned parallel corpora.*

A large and well-aligned parallel corpus plays a key role in the success of a statistical machine translation (SMT) system. Creation of such a corpus is the most critical and time consuming task in developing an SMT system for a given language pair, especially when limited resources are available to form a parallel corpus as it is the case for Persian language.

In this paper, the work towards building a sentence-level aligned English-Persian corpus in a semi-automated manner is presented. The corpus is open ended, which means that it can grow in size and/or more languages can be added to the corpus. The paper is organized as follows: Section 2 describes Persian language and the difficulties in automatic processing of this language. Existing parallel English-Persian corpora are also reviewed in Section 3. Section 4 discusses the use of news articles in parallel corpus construction. Finally, Section 5 describes the development process for the corpus presented in this paper.

2 Persian language

Persian is an Indo-European language, which is the official language of Iran, Afghanistan, Tajikistan, and Uzbekistan. The modern Persian, as written in Iran, is a right-to-left script, which looks like Arabic script but it has its own alphabet and grammatical rules. There are some difficulties in the Persian writing system that make it hard to be processed automatically [2], [3]. This is because of some characteristics of Persian language including the following:

- Character Encoding: In addition to the range of Unicode characters dedicated to Persian, some Arabic characters are sometimes used alternatively. For instance, the letters ک (kaf), and ي (ye) can be expressed by either Persian Unicode encoding (U+06A9 and U+064A) or Arabic Unicode encoding (U+0643 and U+06CC or U+0649). As another example, the Persian letter ه (he) with Unicode encoding (U+0647) is sometimes replaced by Arabic letter ة (Teh Marbuta) with Unicode encoding (U+0629). Such cases usually impose difficulties and errors on automatic Persian language processing systems.

- Tokenization: There are various scripts for writing Persian texts, differing in (a) the style of writing words, (b) using or eliminating spaces within or between words, or (c) using various forms of characters. This makes Persian texts hard to be processed automatically. Tokenization, which is one of the early steps of text processing, therefore becomes a complex and challenging part in Persian language processing [4]-[5].

- Word Order: In Persian, normal sentences are generally structured with a subject-object-verb format. However, sentences may have relatively free word order, referred to as scrambled order. This scrambling characteristic gives Persian a high degree of flexibility for versification and rhyming. This feature, however, makes Persian more difficult to be processed automatically [3], [6].

3 Existing English-Persian parallel corpora

Shiraz test corpus [7] is the first attempt reported on developing English-Persian corpora. This corpus consists of 3,000 Persian sentences collected from a Persian corpus of online material. It was manually translated into English at New Mexico State University to test Shiraz machine translation (MT) system. Some efforts in developing speech-to-speech English-Persian MTs for army force protection and urgent care medical interactions were supported by DARPA[1]. The corpora used in these works have been collected from available corpora in other languages (e.g., English-Iraqi) or in-domain resources such as Medical Phrasebooks and translated manually [8]-[11]. Qasemizadeh, *et al*, made some efforts on building a parallel multilingual corpus for Persian in MULTEXT-East framework [12]. They used the Orwell's 1984 as the main text to construct the corpus. The Persian side of the corpus comprised about 6,606 sentences with about 110,000 tokens. Mohaghegh, *et al*, developed an open corpus from movie subtitles consisting of about 10,000 sentence pairs [13]. Pilevar, *et al*, on the other hand, take advantage of movie subtitles to form the largest parallel English-Persian corpus to date, called TEP [6]. It consists of about 554,000 sentence pairs with about 3 million words in both Persian and English. They, however, admit that movie subtitles contain daily conversations, which are informal and therefore cannot be easily annotated in an automatic manner. This limits the usability of the corpus in Persian natural language processing (NLP) applications. As the last example, European Language Resources Association (ELRA) constructed a corpus (commercially available online) consisting of about 3,500,000 English and Persian words aligned at sentence level to give approximately 100,000 sentences distributed over 50,021 entries. The corpus consists of several different domains, including art, culture, idioms, law, literature, medicine, poetry, politics, proverbs, religion, and science.

4 Using news in parallel corpus construction

There are many resources that can be used to construct parallel corpora (e.g., literary translations, movie subtitles, Wikipedia documents, news, etc.).

- **Literary translations**

Although literary translations have been used in parallel corpus construction, they are less common for machine translation purposes [12]. This is mainly because literary texts are hard to translate even by human translators because of cultural differences between source and target languages. Literary translations are hard to align at sentence level because they usually are translated conceptually. They not only require a thorough knowledge of the source and target languages, but they also need to be able to correctly translate the original feelings and to employ the most appropriate language means in the translation procedure.

- **Movie subtitles**

Another resource for constructing parallel corpora is movie subtitles. As addressed by Pilevar, *et al*, in [6], movie subtitles have various advantages such as large amounts of entries, public availability, similarity of the source and target sentences in length, etc. There is, however, a general disadvantage for using movie subtitles for corpus construction and that is the informality of the sentences used. The problem becomes worse for Persian language noting the fact that for mapping from formal words/phrases to their informal equivalents, there is no mapping table available, nor any tool has been developed for this purpose. Therefore the use of such resources in Persian corpus construction is rarely reported and prone to error.

- **News stories**

To acquire a fairly large parallel corpus that could provide the necessary training data for experiments on statistical machine translation, we chose to mine news stories. There are various advantages in using news stories, such as:

– Large amounts of news stories are written or translated in many languages including Persian and English.

– They are publicly available and can be downloaded freely from a wide variety of online news resources.

– All the news stories in Persian are formal, so it is more appropriate to be used in NLP tasks.

– There is an increasing demand by news agencies for immediate and more accurate machine translations to translate news documents.

According to the above characteristics of news stories, increasing interest has been attracted to develop corpora based on bilingual news stories. For example, Fry built an English-Japanese parallel corpus from RSS news feeds which publish Japanese news stories from English originals [14]. In this work, the links in the Japanese articles to their English equivalents were used to match corresponding document pairs. An English-Japanese comparable corpus was also developed by Utiyama, *et al*, [15]. Nadeau and Foster used Canada Newswire (CNW) news feeds to build a parallel corpus of English-French texts [16]. Huang, *et al*, reported an English-Chinese comparable corpus based on news stories [17]. To the best of our knowledge, Shiraz parallel corpus [2] and the comparable corpus introduced in [18] are the only attempts to build English-Persian parallel corpora from the

[1] Defense Advanced Research Projects Agency

news. Shiraz corpus was initially developed by collecting online news articles from Persian news web sites, extracting a set of sentences varied along syntactic and domain dimensions, and then translating into English by Persian native speakers, manually. Like any other comparable corpus, the corpus introduced in [18] needs more processing (both automatic and manual) to be suitable for statistical machine translation tasks.

In the next section, the procedure of building a new sentence-level parallel corpus for English-Persian machine translation is described.

5 PEN corpus development

Building a sentence-level aligned English-Persian parallel corpus in a semi-automated manner is presented, which can be used in linguistic researches such as statistical machine translation. The corpus development procedure is as follows:

1. A manually-aligned *control corpus* is built comprising 1,200 sentence pairs. This corpus is developed to determine the similarity measures for English-Persian sentences.

2. To build the *main corpus* (PEN), news stories in both Persian and English are downloaded from the website of a multilingual news agency. News story pairs (i.e., news stories in English paired with their translations in Persian) are aligned in document level automatically.

3. Main corpus documents are then aligned in sentence level using the similarity measures obtained in step 1.

4. Quality of sentence alignment procedure is evaluated using Google translator as a reference. In this step, sentence pairs with poor translation quality are first tagged and subsequently removed manually.

Corpus development procedure is illustrated in Fig. 1.

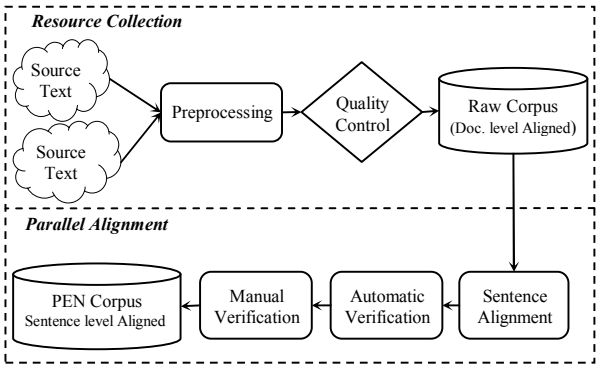


Figure 1 Development procedure for PEN

5.1 Resources

As described earlier, the raw material for constructing the corpus is collected from news stories available online on the website of a news agency. The stories (a total of about 5,700 documents) cover international news, brief news, articles, sports, interviews, etc.

5.2 Control corpus

Fortunately, most of the downloaded documents are already aligned in paragraph level, but in many cases the sentences in the paragraph pairs are not fully aligned. Misalignment of sentences could be classified in two categories:

1. *Sentence disorder*: All sentences in the source paragraph are exactly translated into the target language, but the order of sentences in the target paragraph is different from what appears in the source paragraph.

2. *Sentence mismatch*: (a) Target paragraph is an exact translation for the corresponding source paragraph, but it is expressed in a number of sentences different from what appears in the source paragraph. (b) When translating from one language into the other, some sentences are sometimes preferred to be removed or added.

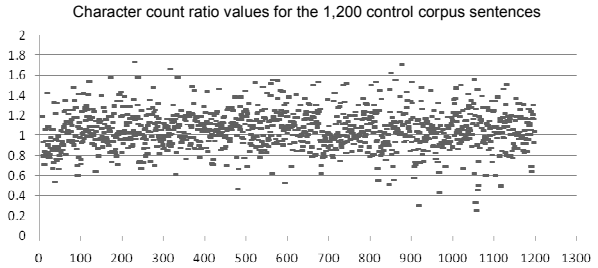
For automatic sentence alignment, some similarity measures are needed. Gale and Church used a simple statistical model of character lengths to align sentences in parallel corpora [19]. They used a normal distribution based on the lengths of sentences in both source and target languages in terms of the number of characters. Then, they specified the model by the mean value and standard deviation parameters of the distribution. In addition to the character-level measure introduced in [19], a similar statistical measure in word level was used for automatic sentence alignment in this work. These measures for the 1,200 control corpus sentence pairs are shown in Fig. 2, mean values and standard deviations of which were used for the alignment of the main corpus sentence pairs.

5.3 Preprocessing

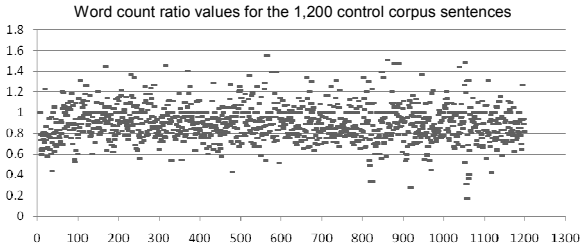
After document-level alignment, the main corpus needs to be prepared for sentence-level alignment. In this step, HTML tags of the aligned documents are removed. A normalization procedure is then performed to correct both character encodings and punctuation mark positions.

5.4 Sentence alignment

The document-pairs are classified into two categories according to their numbers of paragraphs: *Document pairs with different number of paragraphs* are discarded in this step and will be processed manually subsequently. *Document pairs with the same number of paragraphs*, on the other hand, are processed to extract the sentence pairs. Since the documents are selected from a news domain, they are full of



(a)



(b)

Figure 2 Distributions of (a) Character and (b) Word count ratios for the 1,200 sentence pairs of the control corpus

abbreviations[2] and also contain web addresses, both making the detection of sentence boundaries a rather complicated task. To overcome this problem, a sentence boundary detector (SBD) was developed for the extraction of parallel sentences from the corresponding paragraphs. It uses a small set of abbreviations in both Persian and English and some heuristics. This is to discriminate between the dots are used as 'full stops' from those used for other purposes, namely decimalpoint s and the ones used in web addresses (e.g., www.domain_name.com). Trying to process the paragraphs, the SBD first searches for non-full-stop dots and removes them keeping their locations in the text. When paragraph segmentation finishes, the non-full-stop dots are inserted in their original locations. Some sentence pairs may be missed in this step, due to the performance of the SBD module. So, improving the sentence boundary detector especially for Persian language is one of the future works.

Next step is the alignment of the extracted sentences using the two aforementioned similarity measures. In this step, only one-to-one correspondences in text units are used. In other words, a sentence in one language is matched with only one sentence in the other language. Application of this criterion may sometimes cause misalignments or even lead to not finding matching sentence pairs. Both automatic and manual verifications were done to avoid such problems.

[2] The abbreviations that are troublesome in this context are the ones that contain dots in between alphanumeric characters, e.g., U.S.A., U.N., E.U., Prof., and Mr.

5.5 Automatic verification

To improve the quality of the main corpus, Google translator was used to automatically verify the outcome of the sentence alignment procedure described in the preceding section. As illustrated in Fig. 3, automatic alignment verification is performed as follows: For each sentence pair, the English sentence is first translated into Persian using Google translator. Then, stop words of both the Persian sentence (from the corpus) and the output of Google translator are removed. The number of words identically appearing in both of the resulted Persian strings, referred to as the *matching factor*, is used to qualify how well the English-Persian sentence pairs are aligned. Aligned sentence pairs with a matching factor of more than 30% are considered as similar strings. Sentence pairs with the matching factor of less than 30% are tagged to be reviewed manually.

Table 1 shows some statistics of PEN corpus. Examples of the sentence pairs in PEN are presented in Table 2.

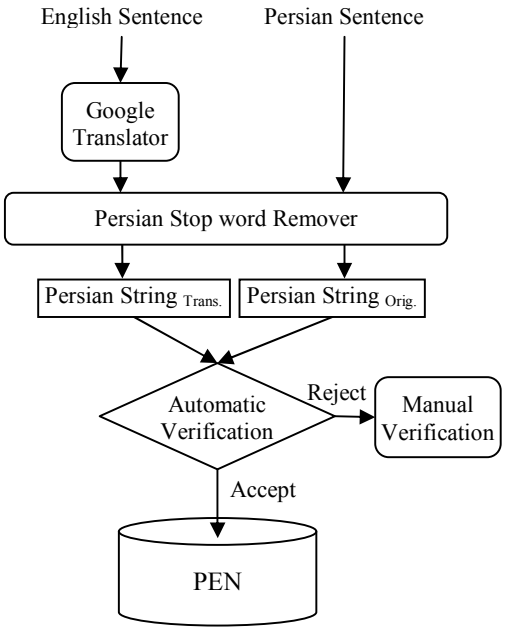


Figure 3. Automatic verification procedure

Table 1. Statistics of PEN

Parameters	English	Persian
Corpus size (in sentences)	30,479	30,479
Corpus size (in words)	567,078	716,089
Corpus size (in characters)	3,571,164	3,399,305
Average sentence length (in words)	18.6	23.5
Average sentence length (in characters)	117.2	111.5

Table 2. Some examples of sentence pairs in PEN

Unfortunately , were just not there yet. متاسفانه ما هنوز به این سطح نرسیده ایم .
In particular he noted that the conditions do not exist for developing major sports in the country. او به طور خاص اشاره کرد که در کشور شرایط رشد ورزش های اصلی مهیا نیست .
The tradition of celebrating the Muslim new year as navruz has been adopted by Uzbekistan , Kyrgyzstan , Azerbaijan , India , Iran , Pakistan and Turkey . برگزاری مراسم جشن سال نو مسلمانان با نام نوروز ، از سوی کشورهای ازبکستان ، قرقیزستان ، آذربایجان ، هند ، ایران ، پاکستان ، و ترکیه برگزیده شده است .
The reasons for delinquency vary . Some blame political and economic instability . Parents and teachers accuse each other of failing to disciple the children . Police say the children learn the behavior by watching adults . دلایل مختلفی برای ارتکاب جرم وجود دارد . برخی بی ثباتی سیاسی و اقتصادی را مقصر می دانند . والدین و آموزگاران یکدیگر را در آموزش ندادن انضباط به بچه ها متهم می کنند . پلیس می گوید که بچه ها رفتارها را با نگاه کردن به بزرگترها یاد می گیرند .

6 Conclusion

In this paper the work towards building a sentence-level aligned English-Persian corpus in a semi-automated manner was presented. The corpus was developed using about 30,000 sentence pairs extracted from about 57,000 news documents available online. The documents cover a wide variety of news domains including sports, politics, interviews, etc. Document-level aligned pairs were first preprocessed. Then, the documents were segmented into sentences and subsequently aligned using two similarity measures. Google translator was used to automatically verify the alignment of sentence pairs. In addition, some manual reviews were done to make PEN more accurate.

7 References

[1] Martin. Wynne. "Developing Linguistic Corpora: a Guide to Good Practice". Oxford: Oxbow Books, 2005.

[2] J. W. Amtrup, H. Mansouri Rad, K. Megerdoomian and R. Zajac. "Persian-English machine translation: An overview of the Shiraz project"; Technical report MCCS-00-319. New Mexico State University, Computing Research Lab, 2000.

[3] K. Megerdoomian. "Unification-Based Persian Morphology"; CICLing 2000, pp. 135-149, Feb 2000.

[4] S. Kiani. "Persian text tokenization and chunking"; 14th International CSI Computer Conference, Tehran, Iran, 2009.

[5] C. Saedi, M. Shamsfard, and Y. Motazedi, "Automatic Translation between English and Persian Texts"; Third International Workshop on Computational Approaches to Arabic-Script based languages (CAASL3), Ottawa, Canada, 2009.

[6] M. T. Pilevar, A. H. Pilevar, and H. Faili, "TEP: Tehran English-Persian Parallel Corpus"; CICLING 2011, Tokyo, Japan, 2011.

[7] R. Zajac, S. Helmreich and K.Megerdoomian, "Black-Box/Glass-Box Evaluation in Shiraz"; Workshop on Machine Translation Evaluation at LREC-2000, Athens, Greece, 2000.

[8] R. S. Belvin, W. May. S. Narayanan, P. Georgiou, S. Ganjavi, "Creation of a Doctor-Patient Dialogue Corpus Using Standardized Patients"; International Conference on Language Resources and Evaluation (LREC),2004 .

[9] E. Ettelaie, S. Gandhe, P. Georgiou, K. Knight, D. Marcu, S. Narayanan, D. Traum, R. Belvin, "Transonics: A Practical Speech-to-Speech Translator for English-Farsi Medical Dialogues"; International Committee on Computational Linguistics and the Association for Computational Linguistics, 2005.

[10] P. G. Georgiou, A. Sethy, J. Shin, S. Narayanan, "An English-Persian Automatic Speech Translator: Recent Developments in Domain Portability and User Modeling"; International Conference on Intelligent Systems and Computing (ISYC), 2006.

[11] N. Bach, M. Eck, P. Charoenpornsawat, T. Köhler, S. Stüker, T. Nguyen, R. Hsiao, A. Waibel, S.Vogel, T. Schultz, A. W Black, "The CMU TransTac 2007 Eyes-free and Hands-free Two-way Speech-to-Speech Translation System"; International Workshop on Spoken Language Translation(IWSLT-2007), Trento, Italy, 2007.

[12] Behrang Qasemizadeh, Saeed Rahimi, "The First Parallel Multilingual Corpus of Persian: Toward a Persian BLARK"; The second workshop on Computational Approaches to Arabic Script-based Languages (CAASL-2), California, USA, 2007.

[13] M. Mohaghegh, A. Sarrafzadeh, "Performance evaluation of various training data in English-Persian Statistical Machine translation"; 10th International Conference on the Statistical Analysis of Textual Data (JADT 2010), Rome, Italy, 2010.

[14] J. Fry, "Assembling a parallel corpus from RSS news feeds"; Workshop on Example-Based Machine Translation, MT Summit X, Phuket, Thailand, 2005.

[15] M. Utiyama and H. Isahara, "Reliable measures for aligning Japanese-English news articles and sentences";

Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, pp. 7-12, Sapporo, Japan, 2003.

[16] D. Nadeau, and G. Foster, "Real-time identification of parallel texts from bilingual newsfeed"; Computational Linguistic in the North-East (CLINE 2004), pp. 21-28, 2004.

[17] D. Huang, L. Zhao, L. Li, and H. Yu, "Mining Large-scale Comparable Corpora from Chinese-English News Collections"; the 23rd International Conference on Computational Linguistics (COLING'10), 2010.

[18] H. Baradaran Hashemi, A. Shakery, and H. Faili. "Creating a Persian-English Comparable Corpus"; Conference on Multilingual and Multimodal Information Access Evaluation (CLEF), pp.27-39, 2010.

[19] W. A Gale, and K. W. Chuch. "A program for aligninig sentences in bilingual corpora"; 2th Annual Meeting of the ACL, 1991.

Post-Logical Verification of Ontology and Lexicons:

The Ontological Semantic Technology Approach

Julia M. Taylor
RiverGlass, Inc &
Purdue University
USA
jtaylor1@riverglassinc.com

Christian F. Hempelmann
RiverGlass, Inc. &
Purdue University
USA
khempelmann@riverglassinc.com

Victor Raskin
Purdue University &
RiverGlass, Inc.
USA
vraskin@purdue.edu

***Abstract*—This paper outlines the methods of practical pre-automatic and automatic verification of evolving lexicons and an ontology used to process natural language meaning, and several approaches that can be taken to speed up the process and decrease the cost of such verification. The main methodological direction of this type of verification is increased automation in conjunction with confirmation by native speakers with basic instructions rather than by fully trained linguist knowledge engineers. It is assumed that ontological and lexical verification based on mathematical logic has been completed prior to the outlined verification.**

***Keywords*—ontological semantic technology, ontology verification, lexicon verification, natural language meaning**

I. Ontological Semantic Technology

Ontological Semantic Technology (OST; see [3, 6, 8]) is a modified and improved version of Ontological Semantics [5] encompasses a theory, methodology, and implementation of a system that attempts to understand natural language text. OST is based on the computer-accessible description of meanings of the words, listed according to their syntactic role in the sentence, taking into account relevant morphological features.

At the core of OST are repositories of world and linguistic knowledge, acquired semi-automatically within the approach, and used to disambiguate the different meanings of words and sentences and to represent them. These repositories, also known as the static knowledge resources, consist of the ontology, containing language-independent concepts, relationships between them; one lexicon per supported language (for example, English), containing word senses anchored in the language-independent ontology, which is used to represent their meaning; the Proper Name Dictionary (PND), which contains names of people, countries, organizations, etc., and their description anchoring them in ontological concepts and interlinking them with other PND entries; and a common sense rules resource.

The process of acquisition of lexicons and ontology was discussed in [5] (Ch. 9) and practiced in prior and subsequent implementations of Ontological Semantics, including the most recent implementation at RiverGlass, Inc. The tools used for acquisition and maintenance of the resources ensure the consistency of the resources to the degree possible, not least in terms of available developer time. Again, this consistency is easiest checked on a purely formal basis, such as syntactic errors (e.g., missing parentheses) in the formal representation language, and on a logical basis, for example, that no lexicon restriction contradicts an ontology restriction or that no ontological concept is more general than its parent.

The lexicon and ontology, together with the PND and common sense rules, are used by the Semantic Text Analyzer (STAn), a software, developed by RiverGlass Inc., that produces Text Meaning Representations (TMRs) from the text that it processes. The format of TMRs conforms to the format and interpretation of the ontology. The processed TMRs are entered into InfoStore, a dynamic knowledge resource of OST, from which information is used for further processing and reasoning.

STAn consists of a collection of modules, each of which fires when a condition that the module is designed to handle is met. STAn reads text, sentence by sentence, by consulting the lexicon and ontology about possible interpretations of each clause in the sentence, combining them, and reporting a ranked number of unambiguous text meaning representations (TMRs) at the grain size provided by the text, as accommodated by the resources. The TMRs that are produced are weighted by the software according to the extent of their fit into each module's interpretation. Thus, it has several major components that could contribute to the incorrect processing of a sentence: the software itself, with each module taking part in it, the lexicon corresponding to the language of the processed text, and the ontology. STAn itself is tested "qualitatively" in weekly plenary team sessions that focus on recent code changes and new linguistic issues that OST has matured to handling. It is also "quantitatively" tested by running a set of sentences before and after each code change and comparing the TMRs from before and after. Any difference is flagged, and the TMR is generated after each module has run to narrow down the set of suspects. This paper, however, will focus on lexicon and ontology verification, assuming the fixed (and bug-free) release of the software.

In general, it is comparatively easy to ensure that lexicons and the ontology are consistent from a logical point of view (e.g., [2] and [1], with a different formalism). The main challenge for post-logical verification is that they may

misrepresent the world as expressed by natural language and lead to incorrect interpretations (cf. also [7]). It is thus crucial to the success of the application to not only check the logical representation of ontology and lexicons, both asserted and inferred, but also how well it works with real text sentences. The tests for the descriptive appropriateness of the resources need to be automated to the highest degree, not least because they have to be complete with respect to every entity in the knowledge resources: every property of every concept and every lexical sense further specifying meaning with these concepts and properties. In addition to automation, the remaining amount of human involvement needs to be shifted from the shoulders of expensive highly trained linguist knowledge engineers to those of cheaper native speakers with minimal instructions. The linguist is rather to focus on the creation of the automated verification methods that the native speakers will interact with. This effort is the focus of the present paper. The motivation and principles of the division of labor are somewhat similar to those in the "Boas" rapid deployment system of Level 2 machine translation for new languages created at an earlier phase of ontological semantics [4].

II. OST Ontology Verification

An OST ontology is a collection of concepts and relationships between them as well as attributes describing the concepts, governed by a subsumption mechanism.

OST ontology can be thought of as a fuzzy ontology, as described in [10]: given a set of objects $\mathcal{D}$, where $\mathcal{D}$ is the disjoint union of $\mathcal{D}$c (concepts) and $\mathcal{D}$d (literals), and given its interpretation function $\mathcal{I}$, for every fuzzy concept B, object x is an element of B with some degree $\mathcal{I}$[B](x) → [0, 1]; for every relation Rel, $\mathcal{I}$[Rel](x, y) ⊆ $\mathcal{D}$c x $\mathcal{D}$c → [0, 1]; for every attribute A; $\mathcal{I}$[A](x, a) ⊆ $\mathcal{D}$c x $\mathcal{D}$d → [0, 1]. We will assume that x∈B if $\mathcal{I}$[B](x) → (0, 1].

With this in mind, the following is true for C and D, using [12]:

$\mathcal{I}$ [C D](x) = max{$\mathcal{I}$ [C](x), $\mathcal{I}$ [D](x)}
$\mathcal{I}$ [and C D](x) = min{$\mathcal{I}$ [C](x), $\mathcal{I}$ [D](x)}
$\mathcal{I}$ [(Rel(D))](x)= $\max_{y \in D}${$\mathcal{I}$ [Rel](y, x)}
$\mathcal{I}$ [(Rel(and C D))](x) = min{$\mathcal{I}$ [Rel(C)](x), $\mathcal{I}$ [Rel(D)](x)}
$\mathcal{I}$ [Rel(C D)](x) = max{$\mathcal{I}$ [Rel(C)](x), $\mathcal{I}$ [Rel(D)](x)}
$\mathcal{I}$ [C(Rel(D))](x) = min{$\mathcal{I}$ [C](x), $\mathcal{I}$ [Rel(D)](x)}
$\mathcal{I}$[C(Rel_1(D))(Rel_2(E))](x)
= min{$\mathcal{I}$[C(Rel_1(D))](x), $\mathcal{I}$[C(Rel_2(E))](x)}

For the construction of the ontology, we follow the subsumption definition of [11]: C subsumes D iff ∀x∈$\mathcal{D}$: $\mathcal{I}$[C](x) ≥ $\mathcal{I}$[D](x).

Since $\mathcal{I}$[C(Rel(D))](x) = min{$\mathcal{I}$[C](x), $\mathcal{I}$[Rel(D)](x)} and $\mathcal{I}$[(C(Rel(D))](x) ≤ $\mathcal{I}$[C](x). Whenever relation Rel is defined with a domain D and range R, if $\mathcal{I}$[C](x) ≤ $\mathcal{I}$[D](x) and $\mathcal{I}$[E](x) ≤ $\mathcal{I}$[R](x), then $\mathcal{I}$[C(Rel(E))](x) is equivalent to min{$\mathcal{I}$[C](x), $\mathcal{I}$[D(Rel(and E R))](x)}, using the axioms above.

These membership degrees should indicate not only logical conformity but also natural language reality. When a request for information is made, the best-suited answer (that containing the highest membership degree) should be given. Thus, for a query "who can drive a car," a response should be "an adult" while "man, woman, engineer," etc., are also perfectly valid from the logical point of view. A native speaker is the best source for testing the appropriateness of the answers given to a natural language question, and thus should be relied on. The same native speaker, however, may not be skilled in logical representation, and is more reliable when asked to confirm or deny something represented in natural language than asking to read through the knowledge representation formalism. We show here how to use this common skill to our advantage in the verification of the ontological knowledge.

A. Best Concept Fit Method

One of the methods of ontological verification is for a software to produce the best concept for a "fill-in the blank" kind of game. For example, a simple sentence is given, such as *I opened the door with zzz*, and the computer has to find the most appropriate concept for *zzz*.

For one version of this method (see [9]), we have created a lexical entry, arbitrarily named *zzz* so as to not conflict with any existing sense, and put ca. 2,000 noun senses, one for every object and event in the ontology, under *zzz*. For every transitive verb sense in our English lexicon (4, 469 in the then current English lexicon of RiverGlass), we take the example sentence that the acquirer entered to illustrate the usage of the verb and replace the noun that is in its direct object slot with *zzz*. This example sentence is then processed by the OST and the best fitting sense of *zzz* is chosen and presented to the native speaker, who decides if the concept chosen is indeed a possible fit, a good fit, the best possible fit, or unacceptable.

A perfect solution was achieved on sentences where either the lexical verb sense (here, *shuck*) or its underlying ontological concepts are very narrowly restricted, like *He shucked the corn*, with the original sentence and the zzz-replacement interpreted by STAn as:

```
(remove(theme(plant-part))
      (agent(human(gender(male))))
      (start-location(grain)))

(remove(theme(plant-part))
      (agent(human(gender(male))))
      (start-location(seed nut grain)))
```

At the opposite end of spectrum lie the sentences that were not interpreted by STAn at a level acceptable for a human judgment. One such sentence was *The engine emitted steam* and the substituted version *The engine emitted zzz*. The unacceptable interpretation of *zzz* was using, as head concepts for the direct object, SHAMPOO, BEER, WINE, and YOGURT. Such misinterpretations are typically caused by the unnecessarily relaxed ontological constraints on some events. In this case, the

event EXUDE (the anchoring concept of this sense of *emit*) has a default theme of GASEOUS-MATERIAL, or LIQUID-MATERIAL, resulting incorrectly in the acceptability of the above substances. On the other hand, the almost 2:1 ratio of the acceptable interpretations suggests that most of the current RiverGlass ontology is reasonably well-constrained.

B. Checking the Grain Size

It is sometimes difficult to select the correct fillers for an event, both for the properties' default position and generic constraints. It is also costly for a knowledge engineer, trained in the formalism, to verify every single change made to the fillers. A less costly solution is to check whether sentences in natural language that corresponds to the default features or generic constraints make sense. Such checks can be performed by humans untrained in the formalism, as long as they are native speakers of the language of the sentences. The Best Concept Fit Method introduced above makes such verification for the default filler. In this section we will outline how to check the generic constraints.

Suppose, there is an event E that has the property P with the filler F. This filler F indicates that both the filler itself and, perhaps, to lesser degrees its descendants, are acceptable arguments to the property P, according to the model of the world that the ontology represents. For example, the event WALK can have LEG as a filler of the INSTRUMENT property, but cannot have LEG's ancestor, LIMB (reflecting that in the world described by this ontology it is not acceptable to walk on ARMS).

To check if the fillers are chosen correctly, a descendant—if it exists—and ancestor of the filler in question, F, are considered. A simple sentence is constructed with the event E that uses property P and its filler F, such as *[All/most] animals walk on legs*. Sentences with descendants and ancestors are also constructed, such as *[All/most] animals walk on hoofed legs* and *[All/most] animals walk on limbs.* All such sentences are given to a native speaker, who has to judge the truthfulness of the knowledge expressed in the sentence. At this point, the grammatical correctness of the sentence is not being considered.

If a sentence with an ancestor concept is found to be true, sentences with siblings of F are generated in a similar manner and are run by a native speaker. In our example, a sentence *[All/most] animals walk on arms* would be generated if *[All/most] animals walk on limbs* was deemed to be true. Whenever sentences with an ancestor of F and all siblings of F pass the native speaker test, a suggestion is sent to the knowledge engineer to rethink (make coarser, i.e., move up to an ancestor) the grain size of the filler for property P of event E, with access to the examples assessed by the native speaker.

If a speaker finds a sentence with a descendant of filler F to be true, as in *[All/most] animals walk on hoofed legs*, several possibilities arise. The first possibility is that both the filler and its descendants were deemed to be correct. Trivially, this can happen if all animals have hooves. This is one question to ask by generating the appropriate sentences, as well as its negative paraphrase such as *There are no animals whose legs are not hoofed*. If the answer is affirmative, the grain size of the filler F should be refined. The second possibility is that, in fact, there are some animals that have no hooves, or, more generically, can be characterized as {CHILD-OF(F)}\HOOFED-LEG. To verify this, a number of sentences is submitted for verification, similar to *Some animals walk on X legs*, where X∈{CHILD-OF(F)}\HOOFED-LEG. If all of them produced a positive response, the grain size is assumed to be chosen correctly.

If a speaker finds a sentence with a descendant not to be true, such as *[All/most] animals walk on hoofed legs,* the positive response is expected to *Some animals walk on hoofed legs*. Otherwise, we may suspect that either the ontology was not constructed properly and HOOFED-LEG should not be listed as a legitimate child of leg, or this particular filler, HOOFED-LEG, must be excluded from an instrument of the event walk. For instance, if there were an animal that had hoofed legs but couldn't use them for walking, similar to some birds having wings but not using them for flying.

As always, there is the possibility of a native speaker's incompetence, the effect of which can be mitigated by using a number of native speakers and accepting only the results that are agreed upon. Since no training is needed for the task, a consistently incompetent native speaker can be easily replaced.

The grain size may need to be checked for the event itself as well: it is possible that E is too generic to describe the property P and its fillers correctly. For the concept WALK, there are some creatures that do walk on ARMS and LEGS. Thus, the choice of INSTRUMENT depends on the choice of AGENT. Practically, WALK has to be split into 2 concepts, those whose AGENTS WALK on just LEGS (whether 4, such as CATS and DOGS or 2 such as HUMANS) or LEGS and ARMS, such as MONKEYS[1]. This can be found out during the verification of siblings of F in the ancestor check. For example, we should get the affirmative answer to *Some animals walk on arms* or *Some animals walk on arms and legs,* while getting the negative answer to *[All/most] animals walk on arms and legs.* A combination of such responses is an indication that a property by itself cannot accurately describe the event without other modification in the ontology.

C. Restricting the Filler

The previous subsection concentrated on a filler and its immediate surroundings, parents, children, siblings. This subsection deals with the bird's eye view, as it were, of each filler.

A great deal of information in the ontology is inherited. This inheritance of properties and their fillers needs to be checked. It should be enough to verify the correctness with a single example of the inherited filler, similar to those outlined in the subsection above.

[1] The example is deliberately chosen to work with English, but, perhaps, not other languages. On the other hand, the front and back limbs of monkeys are sufficiently differentiated physiologically and functionally for the different to be reflected in the ontology.

A concept and its parent normally share a property because the concept was made a child of its parent precisely because a child shares all or most of its parents' fillers. When the filler of the child concept is more restricted than that of the parent, more substantial checks are of use. The most obvious check is a filler that violates the restriction in the child's property filler but was an acceptable filler for the parent's property. Such a check only makes sense if the child filler is significantly more specific than the parent filler or distant from it, and it is possible that a middle-of-the-way filler was overlooked, as shown in Fig. 1.

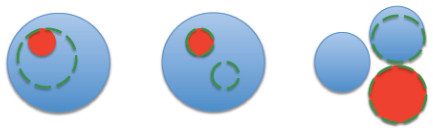


Figure 1. Three cases of parent filler (red and blue), selected child filler (red), correct child filler (dotted green)

There are three distinct cases, as illustrated above: first, when the filler of a parent is a single concept, the selected filler of a child is also a single concept, but the correct selection is a single concept of a considerably coarser size than the selected one; second, when the parent filler is a single concept, as is the one selected for the child, but the correct filler includes an additional concept, thus forming a union of the selected and the missing concept; third, when the parent filler is the union of concepts, one of which is selected for the child, but the correct filler adds a part of another concept. Combinations of these cases can form other possibilities.

Each of them can be tested by randomly selecting several concepts that are distant enough from the selected filler for the child and are subsumed by the parent filler. For each one of such concepts, a sentence involving the child concept for which we are checking the filler is generated, similar to those in the previous subsection. We expect negative judgments on their correctness if the filler is chosen correctly. If the filler is not chosen correctly and a positive response is given for at least one of those test sentences, an incremental expansion of the filler is considered until only negative responses are obtained. The suggestion of the filler concept or a union of concepts is then forwarded to the knowledge engineer.

Let us consider the event LAND-ANIMAL-MOTION, whose INSTRUMENT filler is set at BODY-PART. Its parent is the CHANGE-LOCATION event with the INSTRUMENT filler PHYSICAL-OBJECT. If there were a mistake in this restriction, it would correspond to the first or second cases on Fig 1. Several concepts are semi-randomly selected, typically not on the direct path from BODY-PART to PHYSICAL-OBJECT, and sentences involving LAND-ANIMAL-MOTION event are generated with those selected concepts as its INSTRUMENT fillers. Again, all negative responses are expected.

An additional consideration for the descendant's filler is its combined distribution relative to the parent filler, as shown in Fig. 2. We assume that mistakes such as those described in Fig. 1 have been corrected and that, therefore, the resulting distribution survived the previous verification procedures.

Figure 2. Distribution of children's fillers (red) of a given property in relation to the filler of their parent (red and blue)

If the ontology has a concept C with n of its children, C_1, C_2, …, C_n, none of which inherits C's filler F on property P, but rather restricts F to F_1, F_2, …, F_n, respectively, so that $\bigcup_{i=1}^{n} F_i \subset F$. The question arises then why the balance is not used by any child concept as a filler. This is especially suspicious if the distribution seems to be haphazard, such as in Fig. 2 with a bizarre unfilled gap close to the middle. There is a possibility then that a useful child concept is missing. A reasonable test is generating a text that restricts C to $F \setminus \bigcup_{i=1}^{n} F_i$. The positive answer should be forwarded to the knowledge engineer as a flag for a possible C_{n+1} with F_{n+1} filler of P. The process may have to be repeated until $\bigcup_{i=1}^{n} F_i = F$ or a sufficiently large number of concepts from $F \setminus \bigcup_{i=1}^{n} F_i$ have been tried. If one of the children, C_i, inherits F from C, thus single-handedly excluding the disparity in distribution shown in Fig. 2, it may be useful to remove it from consideration and run this procedure for the remaining children.

III. OST LEXICON

The OST lexicon (one per language) is what connects words of a language with the knowledge of the world captured in the ontology. Each lexical entry (word or phrasal) is represented in all of its senses, focusing on its semantic information (sem-struc) and syntactic environment (syn-struc). The syn-struc indicates the basic syntactic information about the usage of the sense in a sentence. The sem-struc represents the meaning of the sense in terms of ontological concepts and their properties and fillers. Typically, it is anchored in one ontological concept, with its properties and fillers restricted by the usage of the sense. The formal grammar of a typical lexical entry is described in [8].

A. Lexical Gap Detection

The Lexical Gap Detection tool performs the following functions—it:

- runs selected text through STAn and reports, listed alphabetically or by number of tokens or both, in separate files, words for which no senses were found and the senses for words that had them.

- allows for filtering according to word class (e.g., focus on nouns and verbs, ignore named entities).
- presents a good semi-automatic initial step to check coverage of new domains.

B. Sem-struc Verification from Examples in Definitions

For this test, which requires the involvement of trained knowledge engineers, we run for all lexicon senses their example sentences entered by the acquirer to illustrate the usage of the sense through STAn and report the results automatically separated into in four output files:

1. senses that lack an example: example sentences must be added by the acquirers, after which the sense will fall under one of the remaining three classes;
2. senses that did not lead to a TMR: example sentences have to be run individually and the contribution of ontology, lexicon, and STAn modules to the failure assessed;
3. senses that did lead to a TMR, but one that didn't use the sense for which they were supposed to illustrate the usage: same action required as for the previous class;
4. senses that did lead to a TMR using the correct sense.

This semi-automatic method is a good follow-up step to simple lexicon gap detection. In the latter detection, we just report complete lexical gaps, that is, where we find words for which no sense at all is in our resources. The former method, based on example sentence, works better for EVENT senses than OBJECT senses, because if the EVENT is selected incorrectly, it is possible that even a correctly defined OBJECT sense will not be in the TMR due to ontological restrictions of the EVENT. This is especially important for the classes 2. and 3. With this limitation in mind, the first cleaning cycle using this method has led to the following improvement for the RiverGlass lexicon: class 1 down from 41% to 34%, class 2 down from 5% to 4%, class 3 up from 7% to 10%, class 4 up from 47% to 52%. A wide variety of fixes has been applied both to the resources *per se* and the example sentences.

C. Visualizing Virtual Ontology for Lexicon Verification

The ontology is language-independent, and any two entries with identical sem-strucs in any language express the same meaning for the OST system. Thus, within the same language, those senses that have identical sem-strucs are synonyms at the given grain size. An easy check that can be performed by a native speaker is to check the machine-generated lists of words identical in sem-struc regarding their human intuition about the senses' synonymy.

The idea of using native speakers to check word sense relationships against each other can be expanded through the use of a virtual ontology.. With the virtual ontology, each lexical sense is anchored in either a real or a virtual concept, so the ontology has more generations of concepts in between the "hard-coded" ones by virtue of lexical property distinctions (see [8] for a detailed description of the virtual ontology construction). This anchoring allows visualization for each entry in relationship to other entries anchored in the same or related concepts. Thus, not only can we see exact synonyms, but can also look at single-level hypernymy or hyponymy, as well as at siblings.

Let us suppose that we have lexical entries with the corresponding sem-strucs, as follows:

```
(lion-n1
        (sem-struc(lion))
 )
(cub-n1
        (sem-struc(lion(age(value(young)))))
 )
(cat-n1
        (sem-struc(cat))
 )
(kitten-n1
        (sem-struc(cat(age(value(young)))))
 )
(goat-n1
        (sem-struc(goat))
 )
(human-n1
        (sem-struc(human))
 )
(kid-n1
        (sem-struc(human(age(value(young)))))
 )
(kid-n2
        (sem-struc(goat))
 )
```

For each pair of senses *lion-n1/cub-n1, cat-n1/kitten-n1, human-n1/kid-n1*, the visualization of the virtual ontology shows a parent-child node (typically, the first sense being the parent—real—concept, while the second is the—virtual—child concept, since it adds a property of AGE and its filler), as shown in Fig 3. The pair *goat-n1/kid-n2*, on the other hand, is anchored in the same concept. Even without training in the formalism, an observant native speaker should notice an inconsistency in the treatment of adult and young mammals. The fact that an extra sense of *kid* interferes here, should not really make a difference.

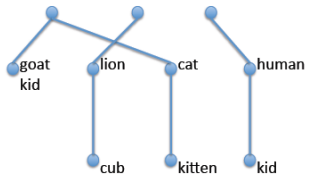


Figure 3. Relative location of some senses according to the virtual ontology

The same method can be utilized in looking for other inconsistencies, such as creating different semantic structures of senses that can be synonyms, missing senses of some lexical entries, and unnecessarily restricted entries.

D. Sentence Verification of Lexical Senses Restrictions

Lexical restriction, similar to ontological restriction, plays a crucial part in resource acquisition. An incorrectly specified

restriction could lead to the rejection of a perfectly meaningful sentence (when the restriction is too narrow) or an interpretation not acceptable to a human (when the restriction is too wide), usually resulting in rather surreal interpretations by the OST system. There are different ways to narrow down an ontologically allowed filler. In the current RiverGlass implementation of OST, the SHOULD-BE-A pseudo-property is used. An example of the sem-struc for the intransitive sense of the verb *walk1*, first without further restriction and then with a—in this case, incorrect—restriction are shown below:

```
(walk-v1
      (sem-struc(walk(agent(value(^$var1)))
)
(walk-v2
      (sem-struc(walk(agent(value(^$var1
                  (should-be-a(sem(human))))))
)
```

The second sense of *walk* restricts the filler of AGENT, specified in the ontology as ANIMAL, to HUMAN. Whenever such a restriction is made, the lexicon verification procedure, analogous to the ontological verification described in II.B and potentially II.C should be performed. Since each sense is necessarily supplied with an example sentence, this example can be used for the verification scenario. Assuming that an entry for *walk* is supplied with the example *He walked in the park*, the following sentences are checked to verify the restriction:

- A human walked in the park. ← selected restriction
- An animal walked in the park. ←hypernym of restriction
- A man walked in the park. ←hyponym of restriction

A positive judgment passed on the acceptability of the second sentence would indicate that the restriction is chosen incorrectly or should be at least questioned.

Interestingly, unlike the cases in the virtual ontology, it is important that verification be done with real concepts, not virtual ones. Thus, because *infant* is anchored in the virtual concept (human(age(value(very-young)))), the negative response to the following sentence cannot falsify the restriction:

- An infant walked in the park. ←hyponym of restriction

The importance of the differentiation between virtual and real concepts is clear here, because if *infant* were anchored in the real concept INFANT, the sentence above would falsify the restriction since infants, while human, cannot walk.

E. Sentence Verification of Unrestricted Lexical Senses

It can be useful to verify the correctness of lexical senses when they are anchored in ontological concepts without any restrictions. While mistakes in unrestricted anchoring should be caught with virtual ontology visualization, the generation of test sentences is fast enough to consider this test.

The test is almost identical to that for restricted lexical entries. The only difference is that the ontological restriction for the entry is tested rather than the lexical one. Thus, in the example of the sense *walk-v1*, the first step is to find what the value of var1 can be according to the ontology. Once the value is established, the hypernym and hyponym tests are run in addition to the selected value tests, as outlined in the previous section.

CONCLUSION

The paper sketches a suit of tests that are useful for ontological and lexical verification in addition to the logic-driven tests that are usually considered in ontological verification. Our opinion is that tests that assess the extent to which the lexicon and ontology conform to the native speaker's intuition are essential for a system that aspires to handle natural language meaning.

REFERENCES

[1] Baader, F., D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, Eds. 2003 *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge: Cambridge University Press.

[2] Guarino, N. 2004. "Toward a formal evaluation of ontology quality," *IEEE Intelligent Systems* 19(4), pp. 78-79.

[3] Hempelmann, C. F., J. M. Taylor, and V. Raskin. 2010 "Application-guided ontological engineering," *International Conference on Artificial Intelligence,* pp. 843-849, Las Vegas, NE.

[4] Nirenburg, S. and V. Raskin. 1998. "Universal grammar and lexis for quick ramp-up of MT systems," *COLING-ACL '98 (36th Annual Meeting of the Association for Computational Linguistics),* vol. II, 975-979.

[5] Nirenburg, S. and V. Raskin. 2004. *Ontological Semantics.* Cambridge MA: MIT Press.

[6] Raskin, V., C. F. Hempelmann, and J. M. Taylor 2010. "Guessing vs. knowing: The two approaches to semantics in natural language processing," *Annual International Conference Dialogue 2010*, pp. 642-650, Moscow, Russia.

[7] Stecher, R. and C. Niederée 2005. "Ontology fitness—supporting ontology quality beyond logical consistency" *Formal Ontologies Meet Industry Workshop*, Castelnuovo del Garda, Italy.

[8] Taylor, J. M., C. F. Hempelmann, and V. Raskin. 2010. "On an automatic acquisition toolbox for ontologies and lexicons in ontological semantics," *International Conference on Artificial Intelligence,* pp. 863-869, Las Vegas, NE.

[9] Taylor, J. M., V. Raskin, and C. F. Hempelmann. 2011. "Towards computational guessing of unknown word meanings: The ontological semantic approach," Cognitive Science Conference, Boston, MA.

[10] Taylor, J. M. and V. Raskin. 2010. "Fuzzy ontology for natural language," 29th *International Conference of the North American Fuzzy Information Processing Society,* Toronto, Canada.

[11] Yen, J. 1991. "Generalizing term subsumption languages to fuzzy logic," *12th Int. Joint Conf. on Artificial Intelligence (IJCAI-91)*, pp. 472-477.

[12] Zadeh, L.A. 1965. "Fuzzy sets," *Inf. & Contr.,* vol. 8, pp. 338-353.

Taxonomy and Evaluation of Markers for Computational Stylistics

Foaad Khosmood[1] **and Robert Levinson**[1]
[1]Department of Computer Science, University of California Santa Cruz, USA

Abstract - *Currently, stylistic analysis of natural language texts is achieved through a wide variety of techniques containing many different algorithms, feature sets and collection methods. Most machine-learning methods rely on feature extraction to model the text and perform classification. But what are the best features for making style based distinctions? While many researchers have developed particular collections of style features – called style markers – no definitive list exists. In this paper we present an organized collection of such style markers with performance data on a diverse set of texts. We show that for each training document, one or more markers exist that can distinguish it from others, providing a basis for a weighted, combined set of markers that outperform any of the individual ones. We examine and categorize 502 style markers, both individually and as a set, and evaluate their performance on several English language text collections.*

Keywords: Computational Stylistics, Style Processing, Natural Language Processing, Machine Learning, Computational Linguistics, Artificial Intelligence

1 Introduction

Researchers have been performing stylistic analysis on text corpora for millennia [1] beginning (in the Western tradition) with authorship attribution efforts in ancient Greece [2]. Nontraditional authorship attribution with internal evidence (only the text itself) is one of the most common applications for stylistics.

Computational methods have been in use in the past decades and have been instrumental in solving several high profile authorship disputes [3]. While these cases and other success stories have shown the power of computational stylistics, the methods used, particularly the style marker selection, have almost always tended to be in support of a traditional theory held by researchers who utilized them [4]. Lack of uniformity in processes, lack of standardization in methods as well as selections of convenience have been among the problems cited in the field [5][6].

We aim to automate a process of statistically deriving the best combination and parameterization of style markers for a given problem. The first step in that process is production of a superset of style markers with detailed definitions to serve as an extensible marker library. We organize these markers into a style marker taxonomy. The superset of markers can be applied to particular problems and relevant markers would be distinguished based on performance.

To find an initial set of markers, we examined many collections in the literature and adopted style markers that were used by the respective authors that could be formalized.

We demonstrate the utility of the taxonomy by applying its markers to our reference corpus and evaluating the markers on attribution performance. We are able to report the top performing markers across eight different English language authorship attribution problems.

In the following sections, we first describe the reference corpus and reasons for choosing it (section 2). In section 3, we state our assumptions with this specific study. In section 4, we describe the taxonomy hierarchy in depth and list current taxonomy marker entries. In Section 5, we run a number of experiments designed to find one or more document discriminators and evaluate the markers in both individual and combined fashion. The results are discussed in section 6. Conclusions are made in section 7.

2 Reference corpus

The Adhoc Authorship Attribution Contest (AAAC) was held in 2004 with many researchers participating [7][8]. The AAAC corpus is particularly well suited for our marker evaluation task for several reasons. The most important reason is that the corpus has purposefully provided significant level of diversity in its many problems.

Documents of different problems differ with each other in genres and text types, as well as document sizes and training/test size ratios, but remain highly uniform within each problem. The corpus is created and prepared by contest organizers and is available to anyone in its original form, allowing for ease of repeatability. The

formatting is machine friendly and in fact, has already been used as an example corpus bundled with the software package JGAAP [9]. The texts are used exactly as distributed in plain text files without the need for any further preparation.

The AAAC corpus is divided into multiple problems. Each problem consists of a set of unlabeled test documents and a set of labeled documents, associated with an author. Usually multiple labeled documents exist per author that can serve collectively as a training corpus, allowing for cross-validation. However, problem H only has one training and one test document for each author.

The types of writing in the problems themselves are diverse. They include student short essays in American English (problems A and B), novels (problem C and G), plays (problems D and E), letters (problem F) and speech transcripts (problem H). Problems I through M are in French, Serbian-Slavonic, Latin and Dutch respectively and are not used in our study mainly because many of the markers used are English specific.

Participants in the AAAC utilized many algorithms each depending on a relatively small set of features extracted from the contest texts [7][8][9]. Each algorithm/feature set/parameter set can be thought of as a "recipe" for authorship attribution. The composition of the recipes, as well as the procedure to apply them however was entirely the work of individual participants based on their own hypotheses.

For our experiment to evaluate markers, we used eight English language problems from the AAAC corpus: problems A, B, C, D, E, F, G and H [7].

3 Assumptions

The fundamental assumption in any feature extraction is that the extracted data is an approximate and representative model of the underlying text, in other words, "style" can be modeled by various markers and associated statistics [8][10].

In order to accomplish the necessary document comparisons, we must ensure that markers remain as universal as possible. Thus, we avoid introducing markers that may not be applicable to some texts, derived from situation-specific corpus comparisons (for example, common word frequencies) or may be direct reflections of the size of a corpus [11].

In general, we can enumerate the following assumptions about the corpora we are considering for stylistics work:

1. A corpus is a collection of documents or just one document.
2. Each document is divided into one or more paragraphs.
3. Each paragraph is divided into sentences (not necessarily a well-formed linguistic definition of sentence).
4. Each sentence consists of words.
5. Each word consists of characters.

Our standard unit of comparison is the corpus. Since a corpus can be a single document for our purposes, we can compare one chapter, one page or even one paragraph (corpus of a single document, single paragraph) against books and collected works, provided our markers are size-independent.

Precise tokenization routines are necessary in order to further specify a uniform way of extracting each of the above units [6]. We hope to standardize these extraction methods as well as the markers themselves. Some of them (for example word tokenizers) can be considered parameters for marker extraction routines as we will describe below. Otherwise, extraction routines will have to be available for examination to make repeatability possible.

4 Taxonomy of style markers

We have developed a taxonomic hierarchy based on previous observations of markers [4][5][8][9][11] [12][13][14]. This hierarchy consists of the following hierarchical elements: *Categories*, *Families*, *Markers*, *Parameters* and *Statistics*. An example illustration for category "Lenghts" is given in Fig. 1 below.

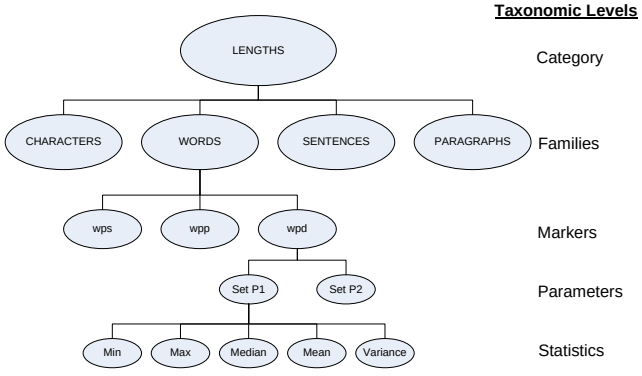


Fig. 1. Partial taxonomic hierarchy for category "Lengths"

4.1 Categories

These are high level themes for the marker collections and subsume many of the marker concepts we encountered in the literature. Examples: lengths, words, n-grams, readability and complexity.

4.2 Families

Families are middle level divisions describing the type of marker in the collection. Examples: characters, words.

4.3 Markers

Markers are base level stylistic events whose presence we are measuring. Example: characters per word, paragraphs per document.

4.4 Parameters

Parameters could be thought of as further sub-division of markers, i.e. variation of markers that may be used simultaneously. In most techniques the parameters correspond to canonizers or pre-processing phase routines which operate on the entire corpus, resulting in the same "parameters" for every marker, as the corpus is modified prior to extraction of any features. This is the way JGAAP[9] handles them and it functions efficiently since only one event set (feature) is being considered at a time.

To accommodate ensemble methods, we must allow for the possibility of using multiple markers with different pre-processing parameters each. Hence, we support separate individual parameterization of each marker. Examples of *parameters* are: unify capitalization, unify numerics (replace with token), exclude common words.

4.5 Statistics

Statistics are metrics used for summarizing marker statistics for the purposes of classification and machine learning. While categories, families, markers and parameters mainly describe a feature event, *statistics* specify how to extract numerical data from the said events. *Statistics* control the footprint of each marker (called marker-instance) in the final feature matrix. For example a *statistics* function for the marker "characters per word" (list of all word lengths in a corpus) could return any of the following:

- A large array of word lengths over the entire corpus (perhaps hundreds of marker instances)
- A single value, the average of all word lengths (1 marker instance)
- A 5-tuple of floats denoting minimum, maximum, mean, median and variance for the array (5 marker instances)

4.6 Types of categories

Categories are the highest level distinctions of the taxonomy. Table I outlines the implemented categories and the families associated with each. Neither the categories nor any of their sub-structure is meant to be a permanent statement on marker style markers or their organization.

Table I. Marker categories

Category	Description	Families
Lengths	Counts and sizes of text features such as sentences and paragraphs.	Characters, words, sentences, paragraphs, syllables, numerics, vowels, punctuation, symbols.
Words	Counts of many categories of words, word frequencies and word lists.	Most frequent words, least frequent words, parts of speech, misspellings, word lists (dictionaries).
N-grams	Counts sequence-based counts of characters, words and parts-of-speech.	Characters, words, parts of speech
Read-ability	Measures presence of well-known readability, sentence complexity and phrases recognized as cliché or poor communication.	Readability (Coleman-Liau, Kinkaid, Flesch Reading, Fog index, ARI, Lix) and Complexity (syntactic depth, parser phrase counts), GNU Diction rules (cliché, rewrite, run-on sentence, superfluous language).
Semantics	Measures having to do with meaning and word senses.	Word Net synset size, synset depth and distance.

4.7 Marker extraction

Due to space restrictions, we are not able to list and discuss every *marker*, *parameter* and *statistic*. We provide a full listing of every marker mentioned in this paper in Table IV. In this section, we choose one category, *lengths*, to demonstrate marker diversity, extraction and values.

The category name "lengths" is meant as in the length of an array. Markers in this category are about counting the occurrences of some distinct feature in terms of another [13], like "words in a sentence" or "vowels in a paragraph." Each of the length markers is an array of integers denoting counts of the phenomenon for which they are named.

For example, let us assume we have a corpus with 2 documents. The first document has 5 paragraphs and the second has 3 paragraphs. The number of paragraphs is determined in accordance with parameter specifications of the marker (4^{th} level of taxonomy hierarchy) which are inputs into the paragraph tokenization routine. For instance, we may want to count a title as a paragraph, or we may not.

The marker output "paragraphs per document" (ppd) is given by an array of two integers representing document length in terms of paragraphs:

ppd (corpus, parameters) = (5,3)

Let us now further assume that the first document's 5 paragraphs consist of 2,5,9,2 and 3 sentences respectively, and the second document's 3 paragraphs consists of 12,3, and 4 sentences respectively, consistent with sentence level parameters. Thus a "sentences per paragraph" (spp) marker is given by an array of 8 integers representing paragraph lengths in terms of sentences:

spp(corpus, parameters) = (2,5,9,2,3,12,3,4)

To compare the markers of two different corpora, which likely have different length arrays, we use statistics, the lowest level of the hierarchy. For example, a *statistics* routine that returns maximum and mean of an array can be used to convert the above *spp* marker array into a tuple of floats.

stats(spp(corpus, parameters)) = (12.0,4.0)

These two values are the final contributions of the *spp* marker to the overall feature matrix of the corpus.

4.8 Other categories

The *words* category, unlike *lengths*, is not concerned with the size or number of all words, but rather the frequency of particular selection words within a corpus or document.

Markers in the "most frequent" family of category *words* are arrays of X decreasing floating point values. One strategy is for X to be chosen as a large number resulting in a large array of floats per marker, and then to summarize that array using statistics resulting in only a few floats per marker. Another strategy is to choose a small number for X, such as 5 or 10 and do not use statistical summaries.

The values in the "least frequent" family are hapax legomena, dis legomena, etc. They are straight integer counts rather than frequencies or ratios. 5-legomena, for example is the number of words in the corpus that occur exactly 5 times each. Part of speech ratios are single floating point values each, thus there is no need for statistics in this family.

The *dictionary* family consists of frequencies of words in a given list L. The size of list L is unspecified and therefore could contain thousands of entries. Statistics may or may not be used on these. "Top100F" family on the other hand is actual frequencies of words, and thus is made up of 100 floating point values.

A popular technique used for stylistics is n-gram analysis [15]. N-grams consist of N successive occurrences of a token type (could be a character, word, part-of-speech or even sentence types). These markers have been successfully used in many authorship attribution problems.

The most important difference between n-grams and other markers is that n-grams are dependent on a meta parameter, N which must be specified to instantiate n-gram routines. Character trigram statistics, for example, could be completely different than bigram ones. To make the taxonomy simpler, we represent n-grams with their generic "n" parameter.

Another important set of indicators are readability indices [16]. Several well-known formulas exist and are reportedly in use to assist essay graders for standardized tests. The particular indices chosen (ARI, Coleman-Liau, Flesch-Kinkaid, Flesch reading ease, Gunning-Fog and SMOG indices) were the ones described in the GNU Style manual, and we refer the reader there for full explanation of each formula [16]. All outputs are single integer or single floating point values.

The *complexity* family markers have array-length outputs similar to those markers in category "Lengths", and thus could be summarized with statistics. "Syntactic depth" is the deepest level of a parse tree achieved by parsing the single sentence in question. Phrase count is a simple count of independent phrases as tagged by the parser, regardless of level or nested status.

The markers in the *semantics* category are mainly derived from Word Net synsets [17] and associated numerical properties with the ontology.

5 Evaluation of markers

One of the principle uses for having a comprehensive taxonomy is that it allows us to experimentally evaluate the markers against particular corpora.

We extract 502 marker instances (marker-parameter-statistics) from all training document collections. This results in a single set of markers for all same-labeled documents within a problem, as well as individual sets for each document. These 502 cover most of the taxonomic markers we presented above with the notable exception of sentence-level complexity statistics (phrase count and syntactic depth).

We used all of the "N-gram" markers with N <=5 and number of top n-grams = 300 for characters, words and parts of speech, meaning the statistics are derived only from the top 300 n-gram events. There was no unification of white space, elimination of numbers and symbols or capitalization for n-gram based markers.

For all word counts, we used alphabetic words only, although for sentence and paragraph lengths, we used all

space separated tokens. For dictionary based operations (stop words, top 1000 English words and top 100 non-stop words) we used a uniform capitalization.

For all array based markers, we used standard *statistics* which reduced all arrays to a 5-tuple of minimum, maximum, mean, median and variance. For single value markers (such as xLegomena, readability indices) as well as for top 100 non stop word frequencies, we did not use any derivative statistics, only the raw frequencies.

We used Python with NLTK tools [18] to extract most of the markers. Part of speech tagging was done by the Stanford Tagger [19], and the LinkGrammar parser [20] was used for some phrase level semantic statistics.

Specifically, we ran two experiments: For the first experiment, we considered each of the 502 marker instances un-weighted, in isolation. We used nearest neighbor with a simple Euclidean distance formula and performed attribution on the problem set based on finding the labeled corpus with the minimum distance.

For the second experiment, we combined all 502 markers using a weight vector. We trained the vector with a fixed number of maximum cycles (200800) for each problem. The resulting weight vectors were used to classify the documents.

6 Experimental results

The eight AAAC problems (A-H) have 187 labeled documents total representing the work of over 50 authors. We classify each document with the given choices in the corresponding problem and thus we derive an "absolute" performance value for each marker.

A problem that has a large number of documents could really dominate the evaluation of the markers if marker performance is measured by absolute number of correct document attributions. Thus, we also calculate the results in terms of problem-relative and marker-relative correct attributions.

The problem-relative results (called adjusted relative performance) are displayed in Fig. 2, along with the absolute performance for each marker. The Problem-relative performance numbers are calculated by considering the percentage of correct attributions per problem, regardless of the how many documents it has.

For example if a problem had only 2 training documents, and only one of them was correctly attributed by marker X, then it would have 50% problem-relative correct attribution for X. If the problem had 10 training documents and still only one was correctly attributed, then X would have 10% attribution score. If one marker was able to correctly attribute every document in the problem it would achieve 100% problem-relative score. Thus, the relative value eliminates the size of the problem as a factor.

Every attribution problem does not have the same degree of difficulty. We are also interested in marker performance relative to other markers per problem. To accomplish this, we rank all the markers from the best performing to the worst for each problem attribution, and calculate difference from the mean in units of standard deviations (z-scores) for each marker, for each problem. This allows us to normalize for the problem difficulty when comparing markers to each other. Table II ranks the top 15 markers and their average z-score performance across all eight problems. For reference, the marker's relative and absolute performances are also listed.

Table II. Marker performance relative to other markers

Marker #	average z-score	rel. perf.	abs. perf.
292	1.65467884	66.29%	55.08%
232	1.47054735	63.72%	52.41%
247	1.34549143	63.88%	49.20%
152	1.29048525	61.84%	50.27%
310	1.24059087	62.47%	49.20%
229	1.22346475	60.98%	47.59%
40	1.21858176	62.66%	44.39%
230	1.21157427	62.12%	48.13%
290	1.21157427	62.12%	48.13%
217	1.21012445	61.59%	48.66%
169	1.18920219	63.19%	44.92%
14	1.17039029	59.15%	44.92%
172	1.15087611	62.98%	44.39%
92	1.13307073	60.92%	48.13%
8	1.12271391	59.68%	45.45%
44	1.11259672	60.44%	45.99%
257	1.11227455	60.73%	47.06%
289	1.11002161	59.69%	48.66%
245	1.10662925	61.31%	44.39%
234	1.08100501	59.34%	47.59%

For the second experiment, we train the weight vector using a modified greedy first-choice Hill Climb algorithm. We also use the k Nearest Neighbor to perform the actual attribution. The results are summarized in Table III. Table III column 1 (left most) designates the problem from the AAAC corpus [4]. Each problem has a number of training documents and a number of possible authors (classes).

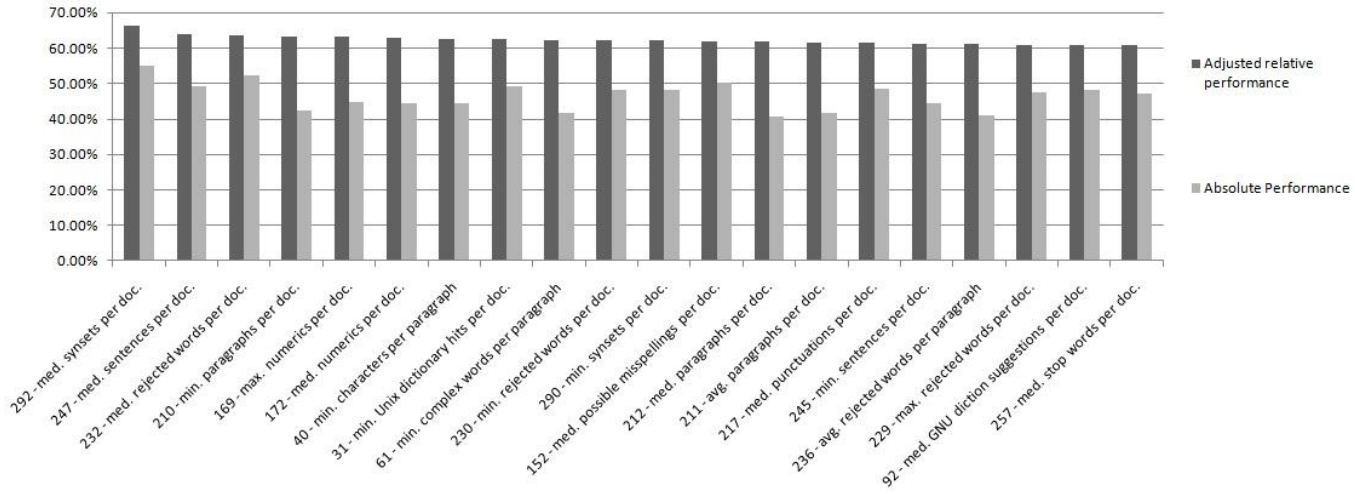


Figure 2. Top performing marker instances by adjusted relative performance.

Columns 2 and 3 of Table IV indicate the best performing individual markers on the particular AAAC problem. For example, the highest number of correct associations that could be performed with a single marker on problem A is 21 documents out of 39 possible or 53.84%. Marker *57* can perform 21 correct attributions and is thus the best independent marker (column 3 from left).

Columns 4 and 5 do the same for the combined approach. Column 4 lists the best attribution results and column 5 lists the top weighted markers in order. We have listed all markers whose weights have grown as a result of training (starting from uniform weight). Therefore, all additional attributions beyond what the top individual marker can do, is due to combining the particular markers in column 5. For problems D, E, G and H, the weighted approach does not add to performance because there exists at least one marker for each of these problems that can by itself categorize 100% of the documents correctly.

For problem H where only three documents needed to be classified, 351 different markers can achieve 100% classification. Another 58 markers can correctly attribute 2 out of the three documents.

Please see the glossary in Table IV for description of the marker instances mentioned in this paper. A full description of all markers used in this study can be found in [21].

7 Conclusions and future work

We have developed an extensible taxonomy of 502 style marker instances in a hierarchical classification. Although many more markers could be added to this collection, either through additional novel markers and extraction routines or production of more variations of the existing markers.

We have conducted experiments to determine the effectiveness of these marker instances.

As can be seen in Fig. 2, and Table II, marker *#292* (median of Word Net synset counts per document) is the best performing marker according to all three individual marker performance measurements (absolute, problem-relative and marker-relative).

Table III. Individual versus combined (weighted) marker performance results

AAAC Problem (docs / authors)	**Best ind. marker results (%)**	**Best ind. mark-ers**	**Best comb. results (%)**	**Top weighted marker instances**
A (39/13)	53.84	*57*	94.87	*92,72,396,57,59, 436,226,298,410, 479,484,244,333, 262,369,288,8*
B (38/13)	55.27	*57*	94.74	*176,39,494,2,13, 228,11,47,19,412 ,33,12,411,417, 14,9,8,459,0,7*
C (17/5)	76.47	*247*	100.00	*79,8,0*
D (12/3)	100.00	*210,211 ,212*	-	-
E (12/3)	100.00	*210,211 ,212*	-	-
F (60/9)	61.67	*14,19*	88.33	*14,19,301*
G (6/2)	100.00	*40,456, 480*	-	-
H (3/3)	100.00	*many**	-	-

**351 markers could distinguish between all three documents in problem H.*

Interestingly, marker *#292* is not highly weighted in any of the combined marker results. Marker *#57* (median of complex words per document) is the highest weighted marker for problems A and B in the combined exercise.

The attribution performance for problems A and B reaches 21 out of 39 and 38 respectively[7]. The training sets for A and B consist of short essays on various topics written by students. For problems with smaller number of documents such as D,E,G and H, there is usually at least one marker that can do perfect classification on the training docs. Thus, the combined method cannot exceed it.

In this paper, we verified two important hypotheses: First, given a large set of individual markers, a few can be found that perform well on a particular set of corpora. Experiment 1 and Table V and VI show the best performing markers in our corpus set. Second, combining markers linearly using a weight vector often performs better than any individual marker on the same data set. However, for smaller sets, combining multiple markers may not be necessary.

We agree with [6] that many problems in authorship attribution studies remain, including lack of standardization and specificity of markers and methods in the literature. Given the severe space limitations in scientific publications, we believe an important step to address some of these problems is to codify markers and extraction routines, complete with full parameterization. This should at least allow the chance to make meaningful and precise comparisons via referencing.

Ongoing work includes further extending the taxonomy and making marker references more descriptive and uniform. We also are applying classification methods based on our markers to different corpora and are maintaining an extensive database of our findings.

Table IIII. Glossary of markers discussed in this paper

Code	**Marker instance**
0	Automated Readability Index (ARI)
2	Flesch-Kinkaid Grade index (FKG)
7	ratio of adjectives to all words
8	ratio of adverbs to all words
9	ratio of other words (other than noun, verb, adjective and adverb) to all words
11	ratio of plural nouns to all nouns
12	ratio of past tense verbs to all verbs
13	SMOG index
14	frequency of most common character
19	max. character bigram frequency
31	avg. frequency of top 300 character 4-grams
33	variance of top 300 character 4-grams
39	max. characters per paragraph
40	min. characters per paragraph
44	max. characters per sentence
47	med. characters per sentence
57	med. complex words per document
59	max. complex words per paragraph
61	avg. complex words per paragraph
72	med. number of unique GNU Diction rules applicable per document
79	max. GNU Diction new sentence suggestions
92	med. total number of diction suggestions per document
152	med. number of possible misspellings per document
169	max. numerics per document
172	med. numerics per document
176	avg. numerics per paragraph
210	min. paragraphs per document
211	avg. paragraphs per document
212	med. paragraphs per document
217	med. punctuations per document
226	avg. punctuations per sentence
228	variance of punctuations per sentence
229	max. rejected words per document
230	min. rejected words per document
232	med. rejected words per document
234	max. rejected words per paragraph
236	avg. rejected words per paragraph
244	max. sentences per document
245	min. sentences per document
257	med. stop words per document
262	med. stop words per paragraph
288	variance of syllables per word
290	min. synsets per document
292	med. synsets per document
298	variance of synsets per paragraph
301	avg. synsets per sentence
310	min. Unix dictionary hits
333	variance of vowels per sentence
369	max. words per paragraph
396	frequency of the word "go"
410	frequency of the word "come"
411	frequency of the word "made"
412	frequency of the word "may"
417	frequency of the word "little"
436	frequency of the word "right"
456	frequency of the word "must"
459	frequency of the word "turn"
479	frequency of the word "animal"
480	frequency of the word "house"
484	max. freq. of the most frequent English word
494	ratio of total hapax legomena to all words

8 References

[1] J. Middleton Murray "The Problem of Style," Oxford University Press, London, 1922, p. 77.

[2] Harold Love, "Attributing Authorship: An Introduction", Cambridge University Press, 2002, pp 14-16

[3] David I. Holmes, "The Evolution of Stylometry in Humanities Scholarship", Literary and Linguistic Computing Volume 13 Issue 3 September 1998.

[4] S. Fish, "What is stylistics and why are they saying such terrible things about it?" in Essays in Modern Stylistics, edited by DC Freeman, Routledge, 1981, pp 53-66.

[5] J. Rudman, "The State of Authorship Attribution Studies: Some Problems and Solutions," Computers and Humanities 31: 351-365, 1998.

[6] J. Rudman "Cherry Picking in Nontraditional Authorship Attribution Studies," CHANCE, vol. 16, No. 2, pp 26-32, 2003.

[7] P. Juola, "Ad-hoc Authorship Attribution Contest" ACH/ALLC 2004, Gothenberg, Sweden, 2004.

[8] P. Juola, Authorship Attribution, NOW Publishers, 200

[9] P. Juola, et. al. JGAAP, a Java-based,modular, program for textual analysis, text categorization, and authorship attribution. http://www.mathcs.duq.edu/~fa05ryan/wiki/index.php/Main_Page , 2009.

[10] J. Walpole "Style as Option," College Composition and Communication, vol. 31, No. 2, Recent Work in Rhetoric: Discourse Theory, Invention, Arrangement, Style, Audience, (May, 1980), pp. 205-212, 1980.

[11] M. Eder, "Does Size Matter? Authorship Attribution, Small Samples, Big Problem," Digital Humanities, London, UK, 2010.

[12] C . DiMarco, "Stylistic Choice in Machine Translation," AMAT, 1994.

[13] F. Khosmod and R. Levinson, "Toward Unification of Attribution Processes and Techniques", ICMLC, Dalian, China, August 2006.

[14] F. Khosmood and R. Levinson, "Toward automated stylistic transformation of natural language text," Digital Humanities, Washington, D.C. , 2009.

[15] V. Kesselj et. Al "N gram-based Author Profiles for Authorship Attribution." Proceedings of the Conference Pacific Association for Computational Linguistics, PACLING'03, Dalhousie University, Halifax, Nova Scotia, Canada, August. 2003.

[16] M. Haardt, GNU *diction(1)* PDF manual, accompanying *diction* version 1.11. (http://www.gnu.org/software/diction/diction.html)

[17] WordNet at Princeton University Cognitive Science Library, http://wordnet.princeton.edu , 2009.

[18] NLTK, The Natural Language Tool Kit, project home page: http://www.nltk.org/, 2009.

[19] Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In *Proceedings of HLT-NAACL 2003*, pp. 252-259.

[20] D. Grinberg, J. Lafferty and D. Sleator, A robust parsing algorithm for link grammars. Carnegie Mellon University Computer Science technical report CMU-CS-95-125, and Proceedings of the Fourth International Workshop on Parsing Technologies, Prague, September, 1995.

[21] 502 Markers used in this paper are listed on the authors' website: http://style.soe.ucsc.edu/node/101 , 2011.

Contrasting Machine Learning Approaches for Microtext Classification

Jeffrey Ellen
jeffrey.ellen@navy.mil
SPAWAR Systems Center Pacific, San Diego, CA, 92117

Abstract - *The goal is classification of microtext: classifying lines of military chat, or posts, which contain items of interest. This paper evaluates non-linear statistical data modeling techniques, and compares with our previous results using several text categorization and feature selection methodologies. The chat posts are examples of 'microtext', or text that is generally very short in length, semi-structured, and characterized by unstructured or informal grammar and language. These three distinct attributes cause different results than traditional long-form free text. In this paper, we further characterize microtext. Highly accurate classification of microtext entries is crucial to facilitate more complex information extraction. Although this study focused specifically on tactical updates via chat, we believe the findings are applicable to content of a similar linguistic structure regardless of domain. This includes other microtext sources such as IM/XMPP, SMS, voice transcriptions, and micro-blogging such as Twitter(tm).*

Keywords: Text Classification; Natural Language Processing; Microtext; Support Vector Machines; Neural Networks; Self Organizing Map

1 Introduction

This paper continues experimentation on machine learning techniques for classification of 'microtext' chat entries. Having established a baseline performance with more traditional and recent text classification methodologies, the goal is to improve on performance through alternate machine learning techniques. These experiments were conducted on unclassified synthetic data that should be characteristic to microtext in any domain. Although microtext was defined in our earlier paper [10], it is necessary to restate it for context. We consider 'microtext' to have three main characteristics that separate it from the traditional documents used in text categorization:

- Individual author contributions are very brief, consisting of as little as a single word, and almost always less than a paragraph. Frequently the contribution is a single sentence or less.
- The grammar used by the authors is generally informal and unstructured, relative to the pertinent domain. The tone is conversational, and frequently unedited therefore errors and abbreviations are more common.
- The text is 'semi-structured' by traditional NLP definitions since it contains some meta-data (in the case of chat it contains a timestamp, a room/channel, and an author) in proportion to some free-text.

The focus of the investigation is determining the similarities and differences, and which alternatives in terms of traditional Natural Language Processing (NLP) and Machine Learning (ML) processes are most applicable.

We focus on military chat rooms, but we believe the results of our research and lessons learned are not specific to the medium of chat, nor the military domain, but are applicable to any situation involving extremely brief 'microtext'. This would include point to point instant messaging via any protocol (such as XMPP), SMS (Short Message Service) common on mobile phones, transcriptions of voice conversations, and micro-blogging which has been popularized by Twitter and similar services.

2 Motivation

It has been established that chat rooms have become a vital medium for war fighters and analysts to communicate information updates [1][12]. Chat rooms support near simultaneous exchange of information amongst a large number of war fighters and analysts, and have advantages over voice communications. Manual monitoring is tedious and expensive. The goal is to increase 'Situational Awareness', which has a particular connotation for military use, but has a corollary for most any industry, business, or community of interest.

Information retrieval and extraction on free text (e.g. long form prose, newswire releases, emails, etc) is a relatively vibrant and burgeoning research area, but we continue to find a lack of studies and experiments on shorter texts, especially where grammar is less formal and abbreviations are more common. We are aware of papers looking at understanding semantic content within these fractured vocabularies, and

even some within microtext[11], but topic extraction is a very different type of task requiring more of a knowledge representation, and also able to leverage more of the existing English language ontologies and references. As electronic communications become more prevalent, we expect 'microtext' sources to become more common, and more important in day-to-day operations within every industry.

We recently completed a survey of existing research on microtext [15]. Although there were some interesting and notable findings, such as O'Connor clustering "statistically unlikely phrases that co-occur" with Tweet Motif[18], or Go and Bhayani performing sentiment analysis using emoticons as noisy labels[19], there was no other research on classification of microtext, and many of the reports could not be generalized more broadly because of assumptions or limitations. For example, basing results on the assumption that a microtext entry contains a single sentence or single sentiment [16]. This technique may work well for a specific domain or application (such as Twitter with its arbitrary and artificial character limit) but the approach does not generalize.

3 Microtext Characterization

Encoding thoughts into an electronic format continues to get easier. At first, capture and encoding was reserved for higher priority items, such as books, contracts, etc. As the internet expanded in parallel with computers becoming more prevalent and less expensive, the barrier was lowered to include essays, newswire articles, etc. The barrier continues to be lowered in at least three dimensions: cost required to encode, accessibility to encoded work, and knowledge required to operate encoding technology.

So text representations of thought and speech are becoming more prevalent daily, and the messages and thoughts encoded tend to become more brief and less formal. The analog equivalent of 'microtext' has always been a part of our modern, communal society, in the form of conversations, journal entries, etc. It's just that when these expressions occurred in spoken dialog, telephone conversations, and paper notebooks, they are not able to be as easily captured, archived, sorted, or discussed. Nor are they able to be rigorously, scientifically analyzed.

Wikis, microblogs, SMS messages, voicemail transcriptions (through ad-supported free consumer technologies such as Google Voice or Jott) are all examples of digital capture of formerly offline analog communication and discussions.

Although they are both encoded as characters, microtext varies in structure and content from long form text, as discussed in the introduction. It is not necessarily the case that microtext is 'noisier' than regular text. From a semantic perspective, the 'signal' in microtext is very strong; the difficulty comes from lack of context, not too much 'noise'.

4 Experimental Corpus

For this study we used a synthetic chat data set which was generated previously. We created the synthetic data to reflect observations of the most common types of phrases observed in secret level chatrooms, with enough generality to conduct the analysis and present the results in a public, unclassified forum. The semantic content and grammatical structure are relatively similar to the actual classified-secret level chat data, so the NLP challenges are replicated even though the information is quite different. The examples in Fig. 1 characterize four rudimentary categories of 'updates', which itself is a category within the general 'item of interest' category. For the purposes of this unclassified study, all items of interest were of the 'update' type. For a more complete description, refer to [10].

4.1 Normalization, Representation, Feature Selection

The training set's vocabulary was normalized by lowering the case of each term, removing stop words, and lemmatizing, which was done with the aid of the NLTK toolkit [3]. It should be noted that lemmatizing was used instead of stemming because we feared a stemmer would incorrectly modify abbreviations and/or irregular terms which appear frequently in chat, particularly military chat. Additionally, a mapping was done for numbers and coordinates to special terms for generality.

FIGURE 1: EXAMPLE SYNTHETIC SHIP UPDATES AND CHAT LINES

Sample Updates

Binary update (possible values are up or down):
`USS Olympia is up`

Numeric update:
`USS Antietam is at (22.299289, 58.086360)`

Class update (possible values are empty/standby/hot):
`USS New Hampshire is standby`

Text update:
`USS Enterprise is at Pearl Harbor HI`

Filler (no update):
`tired ... I did n't sleep much last night`

Sample chat lines (updates are in bold)

(1) wow !!! did n't realize that
(2) no shes not **New Hampshire is empty**
(3) oh .. never knew that standby
(4) hello U20 (16.381914, 140.673367)
(5) poor thing **Kidd is at (26.026418, 164.611238)** PART
(6) whisling dixie up
(7) up 27 male maryland **Ross is at Kings Bay GA** i dont have to know how to read to drive a nice car ..
(8) Yeah , he 's back

For our domain, each individual post is considered a document. Variation of document length (such as a string of posts from the same author) would be an area of future research for another domain.

5 Methods and Performance Measures

A secondary hypothesis that drove the approaches selected for the experiment is that microtext may be 'dense' enough in information theory terms to be able to be treated by methods normally applied to numerical data, whereas our first study focused more on techniques that are generally favored by the natural language processing community. For example, Wang attempted to identify spammers on twitter by completely ignoring the NLP aspect of the tweets completely, and instead treated author contributions as strings of symbols, and compared them using Levenshtein distance, completely ignoring all grammatical and semantic content.[17] Papers such as this one back our secondary hypothesis.

5.1 Feature Selection

We experimented with a number of feature selection methods to reduce the dimensionality of the document vector space for terms. As with our first experiment [10], the methods used include term selection based on document frequency (DF), χ2-test, (CHI), information gain (IG), and mutual information (MI), all of which are detailed in [5]. Furthermore, during feature selection rare terms ($DF \leq 5$) were eliminated, inspired by a feature selection method variation in [6]. This document frequency cut, or df-cut, resulted in a target featureset vocabulary of 762 unique terms. During our experiments we varied the number of terms selected by each method to measure classifier performance.

In addition, for this experiment a synthetic feature set was created containing metadata. For each chat entry, the following features were generated: original entry length, author, timestamp, and chatroom. As this data set was synthetically generated, these features do not provide much use, and provided no statistical impact. Results presented in this paper are using the term selection method described in the first paragraph. In analysis of actual captured data, this meta-data should improve accuracy.

5.2 Performance Metrics

Three standard measures were used to evaluate the text classifiers' performance on the chat data: precision, recall, and F1 measure.

For the multi-class categorization experiments, both the micro-averaged and macro-averaged values of these measures are computed. The micro-averaged calculations give equal weight to each document, while the macro-averaged values give equal weight to each category [9].

6 Experiments

6.1 Experimental Setup

In keeping with the previous study, results are presented for two experiments on the synthetic chat data. The first experiment focused on the binary classification problem of determining whether a post was of tactical interest or not. The second experiment was a 5-class categorization problem (filler, binary, numeric, class-value, or text) focused on determining how well each classifier was able to further determine what kind of naval ship update was contained in the post. Results from the original study using 4 current text classifiers are included for comparison. Based on the premise that the information was dense, various neural network configurations were tried, the results are reported below. Also, various approaches based on Self Organizing Maps (SOMs) were evaluated. This selection was based on two reasons. First, the success of k-NN in the initial experiment might be a clue that other proximity based algorithms would be successful as well. Additionally, research showed that certain SOMs had positive results with NLP-type tasks regarding classification based on document summarization [13].

6.2 Baseline: Previous Results Using Classifiers

In our previous study [10], we selected 4 classifiers that are commonly known for their high performance in text categorization:

- Support Vector Machine (SVM)
- k-Nearest Neighbor (k-NN)
- Rocchio
- Naïve Bayes (NB)

A local implementation was used for the k-NN, Rocchio, and Naïve Bayes classifiers, all of which are detailed in [7]. For SVM, we utilized the freely available SVMlight package [8]. Highlights are presented below in tables I and II.

TABLE I. PERFORMANCE SUMMARY OF BEST BINARY CLASSIFIERS

method	recall	prec.	F1	# features	selection
SVM	0.803	**0.900**	0.849	400	DF
k-NN	**0.846**	0.871	**0.859**	400	DF
Roc.	0.836	0.805	0.820	400	DF
NB	0.740	0.722	0.731	200	CHI

Extended results are available in our previous paper [10].

The baseline for best performance is then roughly 0.85 for F1 for the binary classification experiment, and roughly 0.80 for micro averaged F1, depending on method and number of features.

TABLE II. PERFORMANCE SUMMARY OF BEST MULTI-CLASS CLASSIFIERS

method	# feat.	miR	miP	miF1	maF1
SVM DF	200	0.762	**0.848**	0.803	0.699
k-NN DF	400	**0.954**	0.702	**0.809**	**0.868**
Roc. DF	400	0.835	0.707	0.766	0.768
NB CHI	200	0.403	0.736	0.564	0.437

6.3 Algorithm Description

For this experiment, the results of the three most promising algorithms are presented. Other configurations of these were tried as well, only the best performing configuration is reported upon.

The first is the standard, fully connected, feed forward neural network. Various configurations were used in experimentation, and the standard back propagation with one hidden layer consistently performed the best. Sigmoid activation was used.

The second is a Kohonen SOM with 2 layers. For the binary experiment there were two output neurons, and for the classification experiment there were 5 output neurons. Standard competitive learning strategy was applied.

The third is a Simple Recurrent Network – SRN, with one input layer, one output layer, and one context layer.

6.4 Binary Classification Experiment

For this experiment, we trained each classifier only to identify *interest* posts. Table III shows the experimental results using the best performing feature selection method for each classifier. Note that MI and IG were never the best performing feature method. As with the previous experiment, the values when the feature count was set to 80 were omitted due to the overall poor performance at that number of features.

The SOM classifier using 200 features performed the best in this experiment, with respect to the F_1 measure, as well as overall. The decreased performance at 400 is a likely indicator of overfitting. In comparing with the previous results, the SOM classifier slightly outperformed k-NN.

Our conclusion is that for a binary (2-class) classification task involving microtext, k-NN and SOM are the best candidates.

TABLE III. PERFORMANCE SUMMARY OF BEST BINARY ALGORITHMS

method	recall	prec.	F1	# features	selection
NN	0.786	0.754	0.769	200	CHI
SOM	**0.856**	**0.895**	**0.874**	200	DF
SRN	0.791	0.821	0.812	200	DF
NN	0.799	0.765	0.786	400	DF
SOM	**0.851**	0.878	**0.861**	400	DF
SRN	0.804	**0.919**	0.862	400	CHI

6.5 Multi-class Classification Experiment

For this experiment, each classifier was trained to categorize a post into one of the following five categories: *filler*, *binary*, *numeric*, *class-value*, or *text*. Table IV shows the results for every configuration of each classifier, where *miR* is micro averaged recall, *miP* is micro averaged precision, *miF1* is micro averaged F_1 measure, and *maF1* is macro averaged F_1 measure.

TABLE IV. PERFORMANCE SUMMARY OF BEST MULTI-CLASS ALGORITHMS

method	miR	miP	miF1	maF1	# feat.	selection
NN	0.788	0.745	0.776	0.751	200	DF
SOM	**0.881**	**0.861**	**0.885**	**0.866**	200	DF
SRN	0.816	0.821	0.832	0.841	200	DF
NN	0.844	0.804	0.834	0.81	400	DF
SOM	**0.904**	0.859	**0.877**	**0.861**	400	DF
SRN	0.835	**0.942**	0.766	0.768	400	CHI

Again, SOM is a pretty clear winner, and seems to indicate some overfitting because of the decrease in performance at 400 features. The SRN at 400 features the highest precision score to date, in this experiment or others.

Again, the conclusion is that for n-class classification tasks, that a SOM is the most appropriate classifier, with k-NN seen as a close second.

7 Results at other classification levels

This experiment was repeated on a set of classified data. The size of the data set is significantly smaller because all of the data, both training and experimental, required hand labeling, and a limited amount of labor was available. The results are comparable. The most important aspect is that the algorithms performed the same relative to each other; that is the strongest performing algorithms on the unclassified synthetic data remained the strongest performing algorithms on the classified data sets. This is important, and provides some amount of verification and validity to the original results.

There is a small anomaly in the classified data results, and there are two likely explanations. Most algorithms scored 2-

3% higher, bordering on statistically significant for such a small size. One possible cause is that the classified data set does not contain the same amount of noise as the synthetic set. The more likely explanation is that there was a performance boost due to the precision of hand labeled real world training data, as opposed to the synthetically generated training data used in the unclassified data set.

The other conclusion of note from the different data set is that document frequency continued to be the strongest performing feature selection method. The disparity is not quite as wide as the presented unclassified results.

8 Performance Considerations

The performance of any of the algorithms is relatively pleasing. The majority of the experiments were conducting on a laptop powered by an Intel Core2 Duo (T7500) at 2.20GHz. This correlates to a PassMark CPU Benchmark of 1267[14]. Most algorithms took on the order of 10 minutes to execute a full training cycle, and individual entries were classified in near real time. This is a promising result for the future applications of this algorithm.

9 Conclusions and Future Work

The empirical results of this study suggest that k-NN and SOM are the best options to this point for categorizing our synthetic military chat data. This is significant because our literature search returned few examples of text classification applied to short documents and/or documents containing a large amount of informal language. Our study shows that these two standard text classifiers can be used in these domains.

Our results also suggest that for this domain feature selection based on document frequency tends to perform fairly well, chi-squared feature selection performs slightly worse, while information gain and mutual information tend to perform poorly. This is not unexpected given the concise nature of the source data. This finding holds from the first study.

This preliminary study will help guide the further development of microtext classification on actual military chat.

ACKNOWLEDGMENT

The author thanks the Office of Naval Research and the Space and Naval Warfare Systems Center, Pacific, Science and Technology Initiative for their support of this work. This paper is the work of U.S. Government employees performed in the course of employment and no copyright subsists therein. This paper is approved for public release with an unlimited distribution.

10 References

Number in square brackets (“[ ]”) should cite references to the literature in the main text. List the cited references in numerical order at the very end of your paper (under the heading `References'). Start each referenced paper on a new line (by its number in square brackets).

[1] E. Medina, “Military Textual Analysis and Chat Research”, IEEE-ICSC 2008 Proceedings, Santa Clara CA, August 4-7 2008.

[2] Eric N. Forsyth and Craig H. Martell, “Lexical and Discourse Analysis of Online Chat Dialog”, Proceedings of the First IEEE International Conference on Semantic Computing (ICSC 2007), pp. 19-26, September 2007.

[3] E. Loper and S. Bird, “NLTK: The natural language toolkit”, 2002. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.4.585

[4] G. Salton, Automatic text processing: the transformation, analysis, and retrieval of information by computer. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989

[5] Y. Yang and J. O. Pedersen, “A comparative study on feature selection in text categorization,” in ICML '97: Proceedings of the Fourteenth International Conference on Machine Learning. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997, pp. 412-420.

[6] M. Rogati and Y. Yang, "High-performing feature selection for text classification," in CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management. New York, NY, USA: ACM Press, 2002, pp. 659-661.

[7] Manning, C.D., P. Raghavan, and H. Schutze, Introduction to Information Retrieval, Cambridge University Press, July 2008.

[8] T. Joachims, “Making large-scale support vector machine learning practical”, Advances in kernel methods: support vector learning, pp 169-184.

[9] Z. Bodó “Evaluation of Text Categorization Systems,” April 2007. [Online]. Available: http://cs.ubbcluj.ro/~zbodo/docs/Writings/eval.pdf

[10] K. Dela Rosa and J. Ellen, “Text Classification Methodologies Applied to Microtext in Military Chat”, ICMLA 2009: Proceedings of the Eigth International

Conference on Machine Learning and Applications, IEEE Computer Society, Washington DC, USA, 2009, pp700-714

[11] Topic Detection and Extraction in Chat (Abstract) Paige H. Adams Craig H. Martell pp. 581-588 ICSC 2008

[12] L.T. Duffy, "DoD Collaboration and Chat Systems: Current Status and Way Ahead," Semantic Computing, 2008 IEEE International Conference on , vol., no., pp.573-576, 4-7 Aug. 2008

[13] R. Desjardins, G. Godin, R. Proulx, "A self-organizing map for concept classification in information retrieval", IEEE International Joint Conference on Neural Networks, 2005 (IJCNN '05), pp1570-1574.

[14] Passmark CPU Benchmark Software, Retrieved June 2010 [Online]. Available: http://www.cpubenchmark.net

[15] J. Ellen, "All About Microtext: A Working Definition and a Survey of Current Microtext Research Within Artificial Intelligence and Natural Language Processing" in Proceedings of 3rd International Conference on Agents and Artificial Intelligence (ICAART), Rome, Italy: INSTICC, January 2011, pp. 62–71

[16] A. Pak and P. Paroubek. "Twitter as a corpus for sentiment analysis and opinion mining." In Proceedings of the Seventh Conference on International Language Resources and Evaluation, 2010, pp 1320-1326

[17] A. H Wang, "Don't Follow me – Spam Detection in Twitter", Proceedings of the International Conference on Security and Cryptography (SECRYPT 2010). INSTICC. pp 142-151.

[18] B. O'Connor, M. Krieger, and D. Ahn, "TweetMotif: Exploratory Search and Topic Summarization for Twitter. In Proceedings of the International AAAI Conference on Weblogs and Social Media. Washington, DC, May 2010

[19] Go, A., Bhayani, R., and Huang, L. 2010. Exploiting the Unique Characteristics of Tweets for Sentiment Analysis. Technical Report, Stanford University.

Automatic Programming through Natural Language Compiler

Kalyanasudaram Somasundaram, Harish Swaminathan
Department of CSE, College of Engineering, Guindy, Anna University, Chennai, India

Abstract - *The inherent ambiguity involved in understanding the natural language description by a compiler is the greatest bane to the developers of an automated programming tool. Any average programmer would assimilate a very good logic to solve a problem description but would face problems while coding it. And also it is not possible for programmers to remember the syntax of many programming languages. This paper presents a method to develop a tool that takes in natural language instructions as input and converts them to an intermediate representation. This intermediate representation should help the compiler to convert them into target language with minimal effort. Our aim is to reduce the ambiguity involved in parsing through the natural language instructions of a problem statement and output the corresponding object oriented program and also make any application developer concentrate only on the logic to develop a good algorithm.*

Keywords: *Natural Language Compiler, Automatic Programming through Supervised learning, Syntax Free Programming, Language Level Independence, Unambiguous description of a Problem Statement.*

1 Introduction

Our system- Natural Language Compiler (NLC) runs over a Compiler/Virtual Machine as shown in Figure 1 to convert the Natural language description into a syntactically correct executable code of any language (say JAVA, PYTHON etc).NLC has a lexical analyser, parser and semantic inference engine like any other compiler. The NLC also has a morphological rule engine that allows flexibility in usage of words thereby not restricting the user to limited dictionary words. The lexical analyzer is not the same as the one that we use in a regular compiler. The lexical analyzer will take the tokens that it needs and leaves the rest untouched. So the lexical analyser is similar to SIMFINDER[8] ,it looks for exact match of words in the sentence. If such a match is not available it looks for variation around the existing words with the help of morphological analyser or else it enters into a learning phase.

For example,

Find the sum of ‘a’ and ‘b’.
The tokens needed here are ‘sum’,’a’,’b’.

Semantic analysis is a part of lexical analysis. The parser is responsible for inserting the control structures as a part of the code.

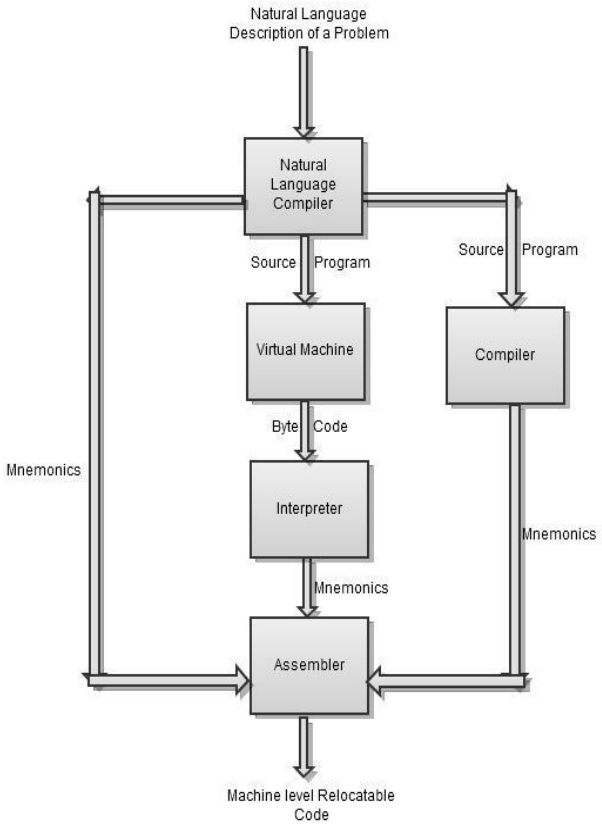


Figure 1- Position of NLC

2 Architecture of NLC

NLC is basically divided into the following phases- Lexical analysis, Semantic Analysis, Syntactic Analysis and Code Generation. Let us discuss each of the phases in detail.

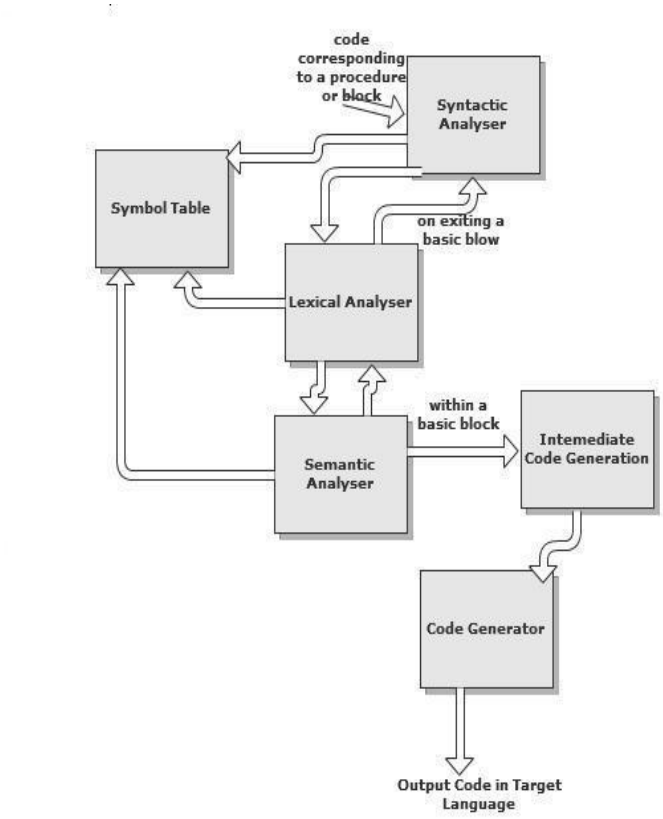


Figure 2-Architecture of NLC

2.1 Lexical Analysis

The input is scanned from left to right until an operator keyword or a synchronisation character is found. The synchronisation character is used to recover from error(s) if the operation goes unrecognized in a statement. A supervised learning method can be used here where the system enters into a dialogue phase with the programmer if operation in the statement goes unrecognised. The programmer can map a word in the sentence with any existing word(s) in the system's knowledge base depicting the operation.

2.2 Semantic Analysis

Usually semantic analysis will take place in a compiler after the syntactic phase. But here since understanding the semantics is the bottleneck it has been placed before the syntactic phase. The lexical analyser pushes operands into the stack. The semantic anlyser determines the operation based on the datatype of the operands.

For example,
equal ‘a’ ‘b’
This triplet can be coded as a==b or a.equals(b) in Java based on whether ‘a’ and ‘b’ are numbers or strings respectively. This ambiguity is resolved in this phase.

To optimise read delays, lexical analyser will buffer a complete statement. A complete statement will start after the synchronising character of the previous statement and continue till the synchronising character of this statement. The syntactic analyser will usually communicate with the lexical analyser through function calls. The call returns back to the syntactic analyser only in the event of change in control flow. In all other cases semantic analyser will work on the output of lexical analyser and form triplets.

The input to our system is a set of natural language atomic instructions i.e. each instruction refers to a single operation. This resembles a traditional three address code with flexible syntax free language. Future research should focus on using compound and complex sentences describing several operations. The work of syntactic analyser will become highly complicated in such a case because to find the order of evaluation syntax tree must be constructed and unambiguous CFG must be framed. This CFG will be compared with intermediate representation realised by the semantic analyser.

2.3 Syntactic Analysis

Each block diagram represents a sequence similar to a basic block[12] and if control transfer occurs at a point p, it is indicated by an edge from block B_i to block B_j. Consider a statement p_i at block B_i and a statement p_j at block B_j such that $i<j$ and the edge has a condition written on it, then it is a conditional jump statement. Similarly if $i > j$ then it is a loop provided there is a single entry to the loop and this edge B_{ij} will constitute the back edge of the loop. From the GUI, the parser has to analyse the flow graphs and produce control structures and punctuators accordingly. The punctuators may be a pair of braces or indentation according to the programming language.

2.4 Code Generation

Code generation technique used in NLC is similar to the code generation technique used in a compiler. To generate code from triplets any existing code generation technique for the compiler can be used if the NLC works at the place of an assembler. But our goal is generating an object oriented code. So for that the system maps action part of each instruction to an operator or method in that language. This mapping is explained in the lower sections.

3 Internal Working of NLC

The working of NLC is actually divided into two different sections which work independently to generate the source code of the description.

3.1 Generating Skeleton Code of a Class

'Draw then code' is the approach that is being followed mostly. From the static class diagrams [3], we can find the class name, attributes, functions and objects of other classes. The skeleton source code of the class can be created from this static class diagram.

3.2 Generating Function Description

As we are done with the class description, we need to get the function description from the user and generate code for the same. Natural language description of a function is a big challenge since the lexical ambiguity and exhaustive vocabulary present in it should be confronted with. While parsing through the descriptions we need to be careful in the following areas:

1).Use atomic sentences. The ambiguity can be resolved if the sentences focus at a particular operation rather than being a compound or complex instructions.

Consider the instruction - Add 'a' and 'b' and multiply it with 'c'. The meaning of 'and' in the first and second context are not the same. If add and multiply were given as separate instructions the chances of ambiguity would be less.

2).A period '.' should be used as a delimiter to mark the end of atomic sentences.

3).Each variable should be enclosed in single quotes.

4).Prepositions cannot be removed as stop words since they have a meaning attached to them.

For example,

Subtract 'a' from 'b' is not same as subtract 'a' and 'b'.

3.2.1 Identification of Operation and Operands

Each keyword should be tagged with a type. This type is used to indicate the type of operation that is whether the operation is of expression type, function type, object methods etc .Expression types are used for relational or arithmetic operations as it needs no function calls. Function types are used for procedural calls in structural programming languages. Thus they will have function names followed by argument. Object method types are used for function calls with the function being invoked by object and '.' operator.The system finds an operation by looking for keywords in the atomic sentence and then forms the semantics based on the operands. The semantics depend on the data type of operands. A same operation say 'equals' will call different operators for numbers and string. The operation will be the verb part of the sentence ,infinitive, or noun form of the verb since only atomic sentences are allowed. Scanning the sentence for a keyword can be optimized by neural networks and training by back propagation[4].

The verb found will not be the exact keyword. It may be morphological variation of the keyword. Morphological differences can be resolved by taking care of suffices either using Porter stemmer algorithm or dictionary access method[5] The operations and operands thus inferred will be sent to the code generator which maps them to the language dependent keywords and variables[6][7]. The code generator will write syntactically correct statement(s) on the output file. Numbers of operands differ depending upon the operation. For example, consider an instruction- increment 'i' . The value with which the variable 'i' should be incremented is missing. Since this information is inadequate the NLC will throw an error.

3.2.2 Dealing with Control Structure

Constructing source code for Control Structures is the most challenging part of our NLC. We can represent the flow of control in a control structure using a Finite State Automata. The developer must draw the possible states with the instructions to be executed in each state[2].Each state in this structure will then be represented with suitable punctuations (curly brace, indentation) to mark the beginning and end of the block(s).

Rules for Drawing State Diagrams:

1). Each state is numbered and has instructions written in it.

2). If a state has two transitions entering into it then that state is the origin of a loop.

3).If a state has a transition without a condition on it then it signifies end of block.

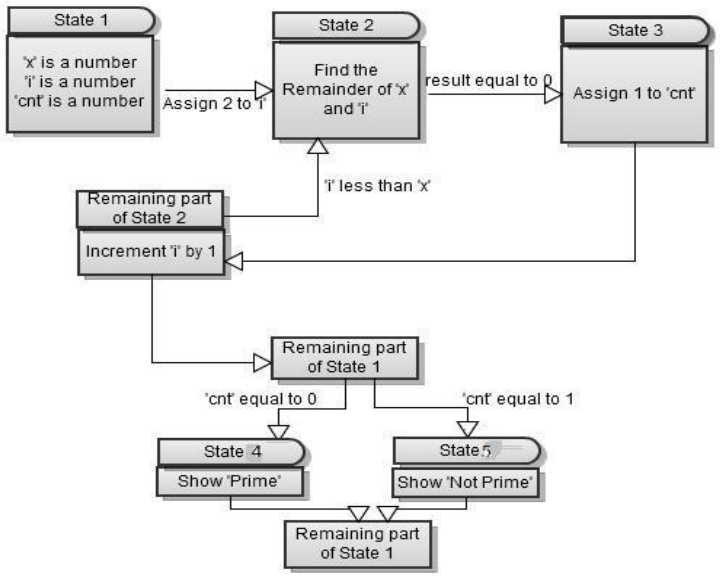


Figure 3-State Diagram to find whether a given number is prime or not

Our NLC system will generate a code similar to the one below for the state diagram in Figure 3:

```
public static void main(String args[ ]){

int x, i,cnt;       //state 1

i=2;                //initial condition of loop

while(i<x){

 /* i<x is the test condition of loop therefore every loop should have two transitions. entering*/

if(x % i == 0 ){

/*x%i is the instruction in state 2 and checking whether it is equal to 0 is the transition to state 3*/

cnt=1;       //instruction in state 3

}             //end of state 3 marked by null transition

i++;         //remaining part of state 2

}            //end of state 2 marked by null transition

             //remaining part of state 1 with no instructions

if(cnt==0){         //transition to state 4 from state-1

System.out.println(prime);

}                    //end of state 4 marked by null transition

else if(cnt==1){ //transition to state 5 from state 1

System.out.println(not Prime);

}                    //end of state 5 marked by null transition

                     //remaining part of state 1

}
```

4 Learning Methodologies in NLC :

Dealing with exhaustive vocabulary is a very complicated job. The only feasible way is to provide a congurable system where the user can add his own keywords when needed. It is not possible to populate all words equivalent to the operations in the warehouse during the initial setup. The system tries to find the operation in each atomic instruction. If it cannot find the operation, then it infers that the user has used a new word to describe some existing operation. If some Learning strategy is not used, then the user will be forced to use restricted set of words for an operation in his sentence(instructions) which would reduce the flexibility of our system. Since flexibility is one of the goals of our system, the system enters into the Learning Phase similar to Learning by Advice strategy. It involves in a dialogue with the user to map the new word in the instruction with the existing words in keyword list. For unsupervised learning, the system can be designed to learn new words by finding their relation with the existing keywords using WordWeb[4].

5 Evaluation of NLC

Some kind of evaluation is needed to measure the real time efficiency of the system. So we compute true positive rate(TPR).

TPR is the ratio of Number of sentences in the instruction that is understood unsupervised to the total number of lines in the instruction.

False Positive Rate(FPR) is the ratio of Number of sentences learnt through dialogue with the programmer to the total number of lines in code.

We tested this with different sets of instruction such as to find Fibonacci series up to N terms, find whether the given number is prime or not, to solve linear equations, to implement substring matching ,etc.

TPR is found to be low initially but later after understanding the vocabulary and the style of the programmer it gets higher and reaches a saturation point. The results are marked in a graph as shown in Figure 4.

5.1 Threats to validity

If the supervised learning methodology fails i.e the knowledge acquired by the system is unstable. This means mapping of an action word(verb or noun form of the verb) changes according to the programmers context every time. Then the graph will have oscillations which is not desired. So learning methodology of this system depends on the programmers' efficiency in providing the knowledge refined.

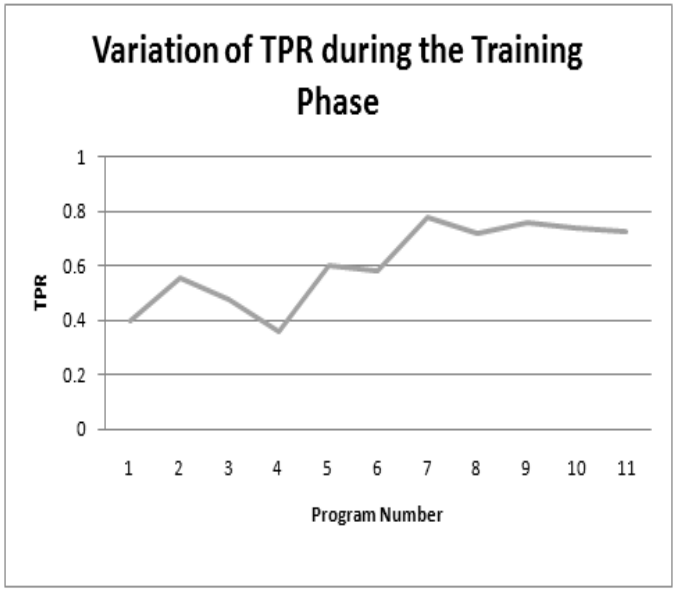


Figure 4 Variation of TPR during the Training Phase

6 Future Areas of Implementation

We can incorporate complete automatic coding by having some templates for frequently used coding parts. We get in the natural language description of the program (probably not as atomic sentences) and match the content of the description with that of the template. This can be done by using topic modeling methodologies like LDA[10] and KL divergence technique. A SIMFINDER[8] can also be used to find the similarities between the descriptions of input instruction and template description. If the input is not in the code repository the system has to generate a newer snippet. But the newer snippet has to be tagged with a description of it. The tagging can be done by an extractive or abstractive summarization[11][9] of input instruction.

7 Conclusion

This paper opens up the possibility of using natural language in Human Computer Interaction.People can concentrate on algorithm and research without minding the platform dependencies. It is time to research on programming language independence. The change that virtual machines have brought can be re-established by this proposed system. Compilers made us forget assembly languages, Virtual machines made us forget the Instruction set Architecture and OS of the system.This proposed system which runs over the VM aims to establish language independency.

8 References :

[1] Hugo Liu, Henry Lieberman - 'Metafor : Visualizing Stories as code', ACM Press San Diego, California, USA. January 2005

[2] Charles Rich, Richard C. Waters – 'Approaches to Automatic programming'-http://www.merl.com Academic Press, 1993

[3] Xing Tan, Gruninger, M – 'Towards Axiomatizing the Semantics of UML Activity Diagrams: A Situation- Calculus Perspective', Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 .

[4] Ronan Collobert Jason Weston – 'A Unified Architecture for Natural Language Processing :deep neural networks with multitask learning', 25th International Conference on Machine Learning 2008

[5] Ando, K. Tsuji, T. Fuketa, M. Aoe,J – 'An efficient dictionary access method for morphological analysis' , Systems, Man, and Cybernetics, 1998.

[6] Javi F. Gorostiza and Miguel A. Salichs - 'Natural Programming of a Social Robot by Dialogs', AAAI Fall Symposium Series 2010.

[7] Charles H. Kellog – 'A natural language compiler for on-line data management' - AFIPS 1968.

[8] Vasileios Hatzivassiloglou, Judith L. Klavans, Melissa L. Holcombe, Regina Barzilay, Min-Yen Kan, and Kathleen R. McKeown – 'SIMFINDER: A Flexible Clustering Tool for Summarization'.- The NAACL Workshop on Automatic Summarisation.

[9] Gabriel Murray and Steve Renals - ,'Detecting Action Items in Meetings' – Machine Learning for Multimodal Interaction 2008

[10] David M. Blei Andrew Y. Ng Michael I. Jordan - 'Latent Dirichlet Allocation', The Journal of Machine Learning Research March 1,2003.

[11] Sarah Rastkar,Gail C.Murphy,Alexander W.j. Bradley - 'Generating Natural Language Summaries For Source Code'

[12] F.E. Allen and J. Cocke - 'A Program Data Flow Analysis Procedure', Communications of the ACM, Volume 19, March 1976

Generating a Sentence from a Thought

W. Faris[1] and K.H. Cheng
Computer Science Department, University of Houston, Houston, Texas, USA

Abstract – *It is desirable for an intelligent program to communicate with humans using a natural language. We have recently developed a program to parse a sentence into a thought using a learned English grammar. Without learning a separate grammar, this paper discusses how we convert the learned grammatical structures into role structures. These role structures can then be used by our algorithm to generate a sentence that reflects the contents of a given thought. Roles define the purpose of grammar terms and link grammatical knowledge to semantic knowledge. Since this linkage separates semantics from grammatical structures completely, the thoughts used in generating a sentence only need to be logical thoughts. Consequently, the creator of a thought may define the thought's content based simply on its semantics and not be concerned with the grammatical details of the natural language.*

Keywords: Grammar, Sentence Generation, Semantic Representation, Natural Language Processing

1 Introduction

One important objective for most artificial intelligence programs is to possess the ability to communicate with humans using a natural language. The communication problem has two major objectives: comprehending the intention of a given sentence, and generating a sentence for a thought that the program wants to express. Recently, we have developed a communication sub-system for the "A Learning Program System" (ALPS) project [1]. The goal of ALPS is to learn all types of knowledge, including those involved in communication. Our system does not use any pre-coded grammatical knowledge, but instead acquires them during program execution. Our system first learns the grammar terms (parts-of-speech) of the English language along with their details, and then uses this grammatical knowledge to parse a given sentence [2]. Subsequently, we have developed solutions to understand declarative sentences [3, 4] including identifying the correct knowledge referenced by various forms of pronouns [5]. In this paper, we discuss how we transform the grammatical knowledge acquired for parsing a sentence into a bidirectional grammar [6] and present how to use that to generate a sentence when given a thought.

Reiter [7] identifies four stages in generating a sentence: content determination, sentence planning, surface generation, and morphology. Content determination involves the formation of an internal representation for the information to be expressed such as feature structures [8]. Sentence planning is the task of mapping the semantic representation to a linguistic one. For instance, it is responsible for identifying the determiner, adjective, and noun when given a semantic representation such as 'the red ball'. Kasper [9] uses sentence planning to develop an interface between a semantic domain and the Penman's sentence generator [10]. However, it requires specialized modifications based on the domain's knowledge base. Surface generation is the process of properly correlating grammatical terms to one another, such as recognizing that an article must precede the noun. The semantic head-driven generation algorithm [11] is an example of a surface generation algorithm. However, it requires the use of a grammar that does not provide a clear separation between the grammatical and the semantic knowledge. In addition, the grammar used is only for sentence generation and requires a separate grammar for parsing. TG/2 [12] is another surface generation tool, but it is limited to a pre-defined grammar. Finally, morphology requires the use of linguistic rules to produce the correct form of a word, examples of which can be found in [13].

One common approach to use a bidirectional grammar is using the same grammatical structure in both parsing and generating a sentence. In other words, when a sentence is being interpreted, parsers identify grammar terms using a set of acceptable sequences and alternatives provided by the grammar; these same grammar terms are used to generate a sentence. The problem is grammars are traditionally designed for parsing, taking text as input. Whereas, in generating a sentence, the input is semantic knowledge. Trying to use the same component in these grammars to generate a sentence has its limitations [14]. Instead, our definition of a bidirectional grammar simply requires that one grammar be used for both purposes, but that it may use different components of the same grammar to accomplish both tasks. The original definition of our grammar uses the grammatical structures to parse a sentence. During the learning of this grammar, we construct knowledge for generating sentences using components known as roles. The role of a grammar term defines the purpose of the term and acts as a bridge between grammatical knowledge and its semantics. Specifically the knowledge created includes role

[1]Research is supported in part by GAANN: Doctoral Training in Computer and Computational Sciences, US Department of Education under grant #P200A070377.

sets, and role sequences. A role set represents the alternative semantic representations of a grammar term, while a role sequence defines a precise way to express a role. Our program compiles this knowledge and sets up the bridge between a grammar term and its associated role. By using these roles, role set, role sequences and bridges, it removes the dependency on the grammatical terms. The process for generating a sentence begins by identifying a sequence of roles that best matches the information stored in the given thought. Each role is then called sequentially to produce a textual representation of that role. Finally, each word is transformed into the right form based on the properties of the knowledge that it represents. Because of the existence of roles, the grammar acquired by our solution does not require special knowledge of the domain. In addition, since roles may be associated to multiple grammar terms, it allows for a simple implementation for a complex grammar. The principles behind creating role sequences ensure the correct grammatical relationships between each term in the sequence, our form of surface generation. As a result, the generated sentence automatically follows the structure taught in the grammar when using the chosen sequence to express the given thought. Our approach has the advantage that no surface generation step is needed at the time of generating a sentence, and thoughts may be created based on its logical meaning instead of the grammatical requirement of the natural language.

Note that we are not interested on how or why the program produces a thought, but on how to generate a sentence according to the contents of a given thought. Currently, our program may use properly constructed thoughts created in three situations. The first situation occurs when a sentence presented to the program by a human user is parsed, creating a thought. We have tested our solution on thoughts such as declarations and questions with various kinds of verbs and pronouns. The second situation occurs when the program is responding to a question. It uses the question thought to create a declaration thought that includes the found answer. The third situation happens during the understanding of a sentence that involves an action. Currently, when certain actions [3] are learned, their effects on various logical objects involved in the action are presented to our program as a sequence of English sentences. The stored effects are individual thoughts created from parsing each sentence. When the program attempts to understand a given sentence that uses that action verb, the thought for each actual effect may be created easily from the prototype effect thought. The actual effect sentence is then generated using our proposed solution. For example, one effect for the action buy is "The buyer gives the price to the seller." Given the sentence "John buys Jack 2 apples from Mary for 4 dollars.", our solution produces the sentence "John gives Mary 4 dollars." as one actual effect.

The rest of this paper is organized as follows. In the next section, we discuss key components of the learned grammar and describe how roles are used to express the intention of a grammar term. In addition, we demonstrate how multiple roles may be combined to form the structure of thoughts expected by our solution. Section 3 discusses the processing needed, at the time of learning the English grammar, to form a bidirectional grammar. This includes building role sets, role sequences, and sequence collections. Section 4 presents the algorithm to generate an English sentence from a given thought. It then describes the algorithm's three major steps in detail: selecting the best-fit role sequence to express the given role, identifying the words representing each role in the selected sequence, and transforming each word into the right form based on the properties of the knowledge that the word represents. Finally, Section 5 concludes the paper and presents some challenges and future objectives of the project.

2 Grammar

The learning of the English grammar is done incrementally in ALPS, i.e., our program first learns a subset of the English grammar and details may then be added to increase the kind and complexity of the sentences that the program can handle. Our program originally uses the learned grammar to parse and understand English sentences. It first learns grammar terms such as sentence, complete subject, verb, noun phrase, and preposition. Each grammar term has several components that define the term and how to use it. Some major components introduced in [2] are structure, role, and rule. The structure of a grammar term defines the exact grammatical format of the term. Two major possibilities for the structure of a grammar term are a sequence and an alternative of grammar terms. A rule specifies a condition that must be satisfied by either the grammar term or its structure. The role of a grammar term defines the purpose of the term. Another grammar term's component introduced in [3, 4] is the control, which is the grammar term in a sequence that carries out an important duty in understanding the semantics of the term. We will show in this paper how to use the same grammar to generate an English sentence when given a thought to express.

A role associated to a grammar term has three important aspects: the grammatical role, the semantic role, and its structure. The grammatical role, identified by *italics*, is the label given to the role depending on the grammatical purpose of the term. For example, the role of the first noun phrase in a sentence is labeled as *subject*. The semantic role, called a logical object, is the label given to the role in relation to its semantic purpose within a higher-level role. For example, the intent of a declarative sentence using a 'be' verb is to define some object of interest. As a result, its role, a declaration, requires two logical objects: definition and object-of-interest. The grammatical *subject* of the sentence serves the semantic purpose of object-of-interest. On the other hand, in a declaration using an action verb, the

same *subject* serves the purpose of actor if the declaration is in active voice. The structure of a role reflects the content of the role, and consists of several related logical objects that represent what the role is expressing. For example, in a thought that reflects an action, the logical objects are actor, act-on object, and act-for object. Consider another example; the aspect-of-an-object structure is one possible structure to realize the role associated to a noun phrase. The knowledge-of-interest referred to by an aspect-of-an-object structure represents an aspect or characteristic of a specific object, the object-of-interest. Consequently, aspect and object-of-interest are the two logical objects in this structure. Given specific values of these logical objects, the knowledge-of-interest may be inferred easily. For instance, if the logical objects aspect and object-of-interest refer to the height concept and Mt. Everest, respectively, then the intent of this role is to express the height of Mt. Everest. One way to express an aspect-of-an-object in English is to use a noun phrase that uses the preposition 'of'. As a result, the grammar term for the preposition 'of' is associated with an aspect-of-an-object role structure. Table 1 shows some role structures and their usage in ALPS. The above example shows a role structure whose logical objects are already provided. However, identifying the logical objects within a structure depends on the associated grammar term. For example, when attached to the preposition 'of' and used in a noun phrase, aspect should be the *simple subject* grammatical role, while object-of-interest should be the *object-of-the-preposition*. On the other hand, a role with this structure could be attached to the possessive noun to express phrases such as 'John's weight'. In this case, the object-of-interest is the grammatical role *possessor*, while the aspect should point to the role of the noun being modified, labeled *term*. In order to create the correct associations, a dynamically generated bridge is created for each instance of the structure linking logical objects to grammatical roles. The structure associated to 'of' will have a bridge that links aspect to *simple subject* and object-of-interest to *object-of-the-preposition*. On the other hand, the bridge for the structure in the second example links aspect to *term* and object-of-interest to *possessor*. By using different bridges, we may apply the same role structure to different grammar terms to express the same knowledge in various ways. The usages of this role structure with its bridges are shown in Figure 1. Note that these bridges link in both directions, one for generating a sentence and the other in creating the thought while parsing a sentence.

It is important to note that the required thought given to our program to generate a sentence is a logical thought, i.e., its content is identified by logical objects instead of grammatical terms. As a result, the creator of the thought only needs to define the logical meaning of the thought, and does not need to be concerned with the grammatical details of the natural language. We will use the thought for the sentence "The next prime number after 7 is 11." as an example. Since this is a declarative sentence, the thought is a declaration. Every declaration depends on its verb role, which is a define role in this example. A define role uses a definition to define an object-of-interest. Therefore, in the declaration, the logical object object-of-interest is 'the next prime number after 7', and the definition is the number 11. The definition object may be represented easily by a whole-object role that contains the number 11. The logical object object-of-interest for the phrase 'the next prime number after 7' may be constructed in a way similar to a declaration. We use a relationship role structure for this expression since it depicts a number related to another number. A relationship role structure is defined by three logical objects: an object-of-interest, a reference-object, and a relation. In this example, 7 is the reference-object, 'after' is the relation, and 'the next prime number' is the object-of-interest. Finally, 'the next prime number' is represented by a role

Table 1. Example role structures in ALPS

Category	Role Structure	Purpose	Logical Objects	Example
Declaration	**Define**	Defines a state, category or property of an object	object-of-interest, definition	Apples are fruit.
	Act	Expresses an action	Actor, act-on, act-for	John gave Jill a gift.
	Possess	Defines the possession of one object by another	possessor, possession	Mary has 2 homes.
Usage	**Aspect-of-an-Object**	Represent an aspect or property of an object	aspect, object-of-interest	His weight
	Relationship	Represent an object in relation to another object	reference-object, object-of-interest	The ball on the table

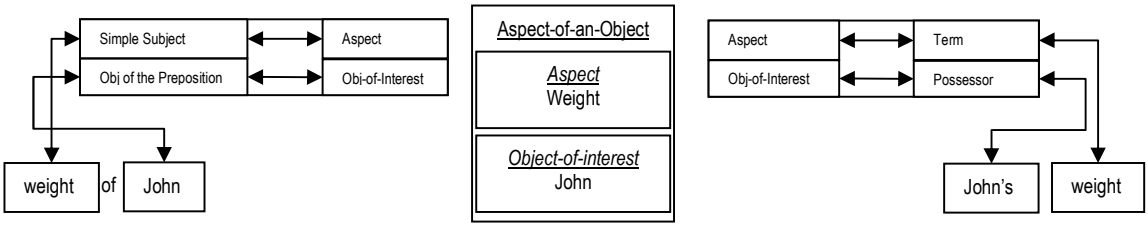


Figure 1. Two bridges of the aspect-of-an-object structure

which contains the knowledge 'prime number', the modifier 'the', which represents a property of 'uniqueness', and the adjective 'next'.

3 Sequence collection

At the time our program learns the grammar, the collection of role sequences for a grammar term is generated according to its grammatical structure. For a grammar term that is defined by a list of alternative grammar terms, its role set is composed of the role structures for the roles associated with its descendants. For example, two alternatives of preposition are 'of' and relational prepositions such as 'above' and 'before'. The structures for their associated roles, aspect-of-an-object and relationship, are added to the role set for preposition. Note that the same grammar may be learned incrementally in many different orders. It is possible that a role is attached to a grammar term that is known to be a descendant of another term. Alternatively, a grammar term already having an associated role is later taught as a descendant of another term. In both cases, the role structure for that role is added to the role set stored at the root grammar term. For instance, suppose define role has already been associated with the 'be' grammar term and verb is initially taught to have two alternatives: action verb and linking verb. When act role is taught to associate with the action verb, it is added to the role set for verb as an alternative. Similarly, define role is also added to the role set for verb when 'be' is taught as the child alternative term of linking verb. As a result, the role set for verb contains an act and define role structures as its alternatives.

For a grammar term whose grammatical structure is a sequence, a role sequence is created based on its order and occurrence. Each item in the role sequence is a grammatical role, with a pointer to the role set of the corresponding grammar term. When a new role structure is added to the role set, it will be available dynamically to any role sequence that contains the role set. For instance, a noun phrase may be defined by the sequence [nominal, prepositional phrase], where nominal is compulsory and has the *simple subject* role, and prepositional phrase is optional and defined in turn by the sequence [preposition, nominal] with this nominal having the *object-of-the-preposition* role. The corresponding role sequence created for noun phrase is [*simple subject*, *prepositional*, *object-of-the-preposition*] with the first item compulsory while the last two items optional. Unlike roles and role sets that are attached with grammar terms, a role sequence is attached to each role structure in the role set of the control. Once a new role sequence is attached to a role structure, that structure may now be expressed by the originating grammatical sequence. For example, the role sequence for noun phrase would be attached to each alternative role structure in the role set of preposition such as aspect-of-an-object and relationship. Recall that a role structure such as aspect-of-an-object may be used by multiple terms each having a different role sequence, the collection of these role sequences is called the sequence collection for that role structure. Any role sequence found in its sequence collection may be used to express the content of the role structure. For example, since the aspect-of-an-object structure may also be used by possessive noun, the role sequence for possessive noun, [*possessor*, *term*], is also added to the sequence collection for the aspect-of-an-object role structure. Figure 2 shows the sequence structures that may be taught for a noun phrase and the corresponding role sequences created for the aspect-of-an-object role structure. These dual structures allow one grammar to both parse and generate a sentence using the grammatical structures and the role sequences, respectively. The next section will show how to choose from a sequence collection a specific role sequence to produce a statement that properly reflects the contents of the role.

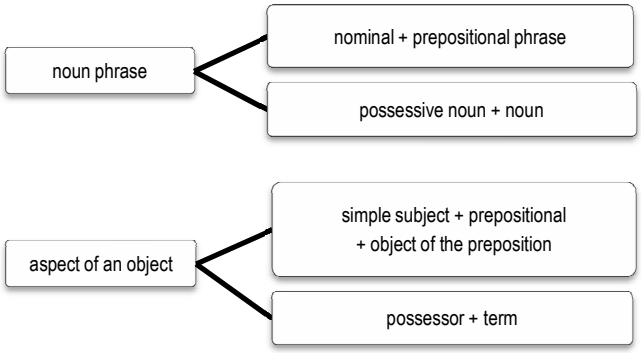


Figure 2. An example dual structure of grammatical structures and role sequences.

Finally, when a sequence is taught as a grammatical structure for a specific kind of a grammar term, the role structures in the role set of the control are not the appropriate place to store the generated role sequence. For example, the control of a sequence applicable to many different kinds of sentences is the verb. Consequently, the generated role sequence [*subject*, *verb*, *predicate*] is associated to each role structure in the role set for verb, in particular, define. On the other hand, decision question is a kind of sentence that has a special sequence structure, which only applies to it, yet it has the same control, the verb. If the generated role sequence [*verb*, *subject*, *predicate*] is also stored in each role structure of the role set of the control, such as define, then two errors may occur. Either the special structure is used erroneously to generate sentences of other kinds or decision questions may be generated erroneously by the generic structure. To prevent these errors, when a sequence is taught for a kind of a grammar term, the generated role sequence is added to the sequence collection of that kind's unique role.

4 Sentence generation

Given that a thought is a collection of roles, each representing a unique semantic element to be expressed, the task of producing an English sentence to reflect that thought involves expressing the intent of each internal role. The intent of each role is represented by the structure of the role. Our algorithm to generate a sentence consists of two major steps: generate and make. The generate step, called by every role structure needed to be expressed, is further subdivided into two steps: select and generateList. The select step is our version of sentence planning, as it selects the role sequence from the sequence collection that best fits the given thought or role. In general, the sequence collection is retrieved from the role to be expressed. However, for thoughts, the location of the sequence collection depends on its kind. The sequence that best fits is the one that expresses all the information contained within the role. The generateList step assembles a list of word-units according to the chosen sequence of roles. The make step performs the duty of morphology. It selects the correct form of the word based on the properties stored in each word unit, and applies any syntactic rules needed to generate a complete sentence.

Recall that the sequence collection of a role contains multiple role sequences that may be used to express what the role represents. However, not all sequences apply to the given role, and some may be more desirable than others. The select step selects the sequence that best fits the given role. For example, suppose declaration has the following sequences in its collection: [*subject*, *verb*, *predicate*], [*subject*, *verb*, *direct object*], and [*subject*, *verb*, *indirect object*, *direct object*]. Now given a thought that expresses the act of a boy selling lemonade to his neighbors, although the first sequence may be used to express a declaration to define a subject, it is not suitable to express a declaration about an action. The second sequence is for an action, but does not express all the information stored in the thought; specifically, the indirect object representing the neighbors cannot be expressed. The last sequence is the best fit for this thought and should be the one selected to generate the sentence. We make use of an algorithm, checkAvailability, to eliminate all sequences that do not match the given role by determining the availability of each element in the role sequence. A role sequence is suitable to express a given role if the given role contains all the logical objects required by the role sequence. If a required element is missing, the role sequence cannot be used and is eliminated. Recall that an element in a sequence can be a single role, a role set, or another role sequence. If it is a role sequence, the checkAvailability algorithm is called recursively on this inner sequence. If the element is a role set, the checkAvailability function is called for each role structure in the set. As long as one alternative is available, the entire element is considered available. If the element is a single role, it needs a bridge to determine if the element exists in the given role. The reason is grammatical roles are used in role sequences, but roles to be expressed contain logical objects instead. The correct bridge may be determined by either the given role or the role of the control term. Continuing with our example thought about an action, since the control of a sentence is the verb and the role for an action verb is the act role, the bridge for an act role is used. Using this bridge, the program knows that the *subject* maps to the actor, the *direct object* to the act-on object and the *indirect object* to the act-for object. As shown in Figure 3, these logical objects are all in the given thought and can be found, indicating that the role sequence is valid.

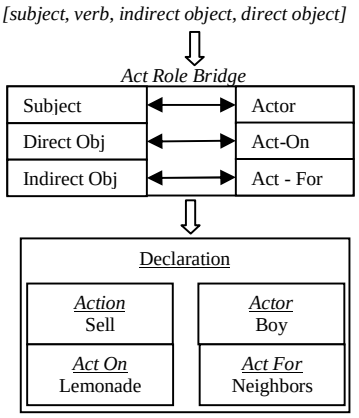


Figure 3. Matching a role sequence with logical objects

Once a logical object is found, its corresponding object in the role is marked as used to prevent another element from using it. For instance, a sentence may have multiple prepositional phrases. If not marked, the same role may be used by each prepositional phrase, resulting in duplicate phrases. If the logical object exists in the given role, the role element is available. However, if an element in a role sequence is not available and its occurrence is compulsory, then the role sequence is eliminated. On the other hand, if its occurrence is optional, it is still possible to use the role sequence and so it is not eliminated. For example, the first sequence in the sequence collection for sentence is not acceptable because the compulsory predicate cannot be mapped to any object in the act role. After removing all invalid sequences, it is still possible to have a set of valid alternatives. In our example, two valid sequences may still be used: [*subject*, *verb*, *direct object*], and [*subject*, *verb*, *indirect object*, *direct object*]. The latter sequence will be chosen since it expresses the most information from the given thought.

After selecting the best sequence of roles to express the thought, the generateList step will work on each role according to its order in the sequence. There are two categories of roles: composite and non-composite. A composite role such as aspect-of-an-object contains multiple sub-roles representing the logical objects of the role, while a

non-composite role is atomic. If the role is composite, the generate function is called on that role and the process of selecting a sequence and composing a word unit list is handled at the level of the composite role. If the role is non-composite, a word-unit is created by extracting the content of the role including the name of the knowledge and any properties associated with that knowledge. For example, a role representing the person John would produce a word-unit containing 'John' and the properties singular, third-person, male, and unique. Note that some of these properties may be stored within the thought role such as the tense of the sentence where the tense property of a verb may be obtained. The object-oriented nature of ALPS allows for each type of non-composite role to have its own unique function to determine what properties are needed and where to locate them.

The make step selects the correct form of the word based on the properties stored in each word unit, and applies any syntactic rules needed to generate a grammatically correct sentence. Currently, we have focused on properties such as case for nouns, and both case and tense for verbs. Two categories of morphology rules: regular and irregular, are used in English to distinguish the various values of a property for a word. For regular words, they follow a set of rules that determine how affixes are added to root words in order to express a property. This allows the same affix to be applied to a large number of words. For example, the suffix –s can be applied to nouns to form the plural of the word. Our program is taught a set of rules that is used to understand a word based on its root form and any affixes found, and this set of rules is used to produce the correct word when given the property. Each morphology rule is taught with a condition to determine if that rule is applicable to the current word, the transformation to apply to the word, and the new property that the word has. For instance, one rule for forming the plural of a word could have condition that if the word ends in y, then transform the word by replacing the -y with an –ies. For irregular words, the difference between the root word and its varying forms does not adhere to any consistent rules. For instance, even though the plural of house is houses, the plural of mouse is mice but not mouses, and dice is the plural form of die. It is estimated that there are around 250 irregular verbs alone in the English language [15], and each special case needs to be learned individually. In our program, specialized lexicons identified by their implied properties are learned for use to map a word between its base form and its irregular form. For instance, a plural lexicon that contains a set of words and their irregular plural forms would contain pairings such as goose/geese, ox/oxen, and radius/radii.

A word unit that does not contain any properties indicates the word is already in the correct form and no conversion is needed. If it has properties, the make step first tries to convert the base form of the stored word into its irregular form. For each property in the word unit, the corresponding lexicon is looked up. If the word in question is found within that lexicon, this indicates that the property can be applied and the irregular form found in the lexicon is used in the final sentence. If all properties have been tried and an irregular form is not found, the make step will try morphology rules for regular words. Finally, whether or not a word has been transformed, all word-units are tested for syntactic rules on capitalization. For example, the first word in a sentence and proper nouns are capitalized.

One final complication of our solution arises from the fact that not all grammar terms have a role associated with them, yet that grammar term still needs to be represented in a role sequence in order to produce a proper expression. For example in our grammar, the grammar term punctuation is a part of a sentence, but does not have an associated role. In this instance, the role sequences for a sentence will have any empty role set reflecting the punctuation. In order to produce a valid sentence, compulsory terms without roles use rules to indicate the proper way of generating the correct output. For example, two rules taught for punctuation are that a declarative sentence must end in a period and an interrogative sentence must end with a question mark. As a result, whenever a role set is empty, the algorithm will use the applicable rule to determine the correct word unit to complete the sequence.

5 Conclusion

In future versions of our sentence generating algorithms, we aim to tackle subtle problems in sentence selection, such as generating passive voice sentences, and in morphology, such as property prioritizing. When deciding to express a thought in the passive voice, a variety of bridges may be used depending on the structure of the passive voice. For instance, take an original thought expressing that John gave a gift to Mary. In the thought, John is identified by the logical object actor, the gift is the act-on object, and Mary is the act-for or beneficiary. This thought could be expressed in several passive voice sentences such as "A gift was given to Mary." or "Mary was given a gift". To express the first case, the bridge would need to link *subject* to the act-on object and in the second; the bridge would need to link *subject* to the act-for object. As a result, a method needs to be developed to determine which object is the focus of the sentence and thus choose the correct bridge to properly express the sentence in that manner. The issue of property prioritizing occurs when certain modalities, or helping verbs, are presented. In certain cases, a property of a word may not need to be applied. For instance, verbs generally take the same case as the subject in a sentence. However, most verbs do not differentiate between singular and plural forms when preceded by a helping verb: "He may run a marathon.", rather than "He may runs a marathon." Similarly, the 'be' verb takes the root form, ignoring all properties, when it is preceded by a helping verb: "John could be a civil engineer." instead of "John could is a civil

engineer." The distinction of when properties can and should be applied is one that needs to be taught through rules that are more sophisticated.

Traditional bidirectional grammars rely on the same structure that was used in parsing a sentence to produce a sentence. This limits the flexibility and usefulness of the learned grammar. Instead, our program converts the learned grammar structure into a secondary type of knowledge that may be used to generate a sentence. This is accomplished by combining roles into sets and sequences to create a parallel structure based solely on the roles instead of grammar term. The purpose of a role set is to collect the alternative semantic representations of a grammar term, while role sequences define the various ways the intent of a role can be expressed. A role can then select an applicable role sequence from its collection to create a list of word units that expresses its contents. These word unit lists then propagate up to the thought to produce a list that expresses the entire thought. Finally, by applying properties gathered from the roles and the thought, the proper form of each word may be produced to create the final output sentence.

Learning grammar incrementally allows more complex structures to be added as desired, in turn expanding the type of sentences that can be parsed and generated. By using roles, with a clear separation between grammatical and semantic purposes, along with the use of the proper bridge, our approach has the advantage that the same logical structure may be expressed in multiple ways in the natural language. The principles behind creating role sequences ensure the correct grammatical relationships between each term in a sequence. It has the advantage that no surface generation step is needed at the time of generating a sentence, and thoughts may be created based on its logical meaning instead of the grammatical requirement of the natural language.

6 References

[1] K. Cheng. "An Object-Oriented Approach to Machine Learning"; International Conference on Artificial Intelligence, 487-492, 2000.

[2] W. Faris & K. Cheng. "An Object-Oriented Approach in Representing the English Grammar and Parsing"; International Conference on Artificial Intelligence, 325-331, 2008.

[3] E. Ahn, W. Faris, & K. Cheng. "Recognizing the Effects caused by an Action in a Declarative Sentence"; International Conference on Artificial Intelligence, 149-155, 2009.

[4] W. Faris & K. Cheng. "Understanding and Executing a Declarative Sentence involving a forms-of-be Verb"; IEEE International Conference on Systems, Man, and Cybernetics, 1695-1700, 2009.

[5] W. Faris & K. Cheng. "Understanding Pronouns"; International Conference on Artificial Intelligence, 850-856, 2010.

[6] D. Appelt. "Bidirectional grammars and the design of natural language generation systems"; In Proceedings of the 1987 workshop on Theoretical issues in natural language processing, Association for Computational Linguistics, Stroudsburg, PA, USA, 206-212, 1987.

[7] E. Reiter. "Has a consensus NL generation architecture appeared, and is it psycholinguistically plausible?"; In Proceedings of the Seventh International Workshop on Natural Language Generation, Association for Computational Linguistics, Stroudsburg, PA, USA, 163-170, 1994.

[8] S. Shieber. "An introduction to unification-based approaches to grammar" , CSLI Lecture Notes , 4 , Stanford Univ. 1986.

[9] R. Kasper. "A flexible interface for linking applications to Penman's sentence generator"; In Proceedings of the workshop on Speech and Natural Language, Association for Computational Linguistics, Stroudsburg, PA, USA, 153-158, 1989.

[10] E. Hovy. "The current status of the Penman language generation system"; Proceedings of the workshop on Speech and Natural Language, Association for Computational Linguistics, Stroudsburg, PA, 1989.

[11] S. Shieber, G. van Noord, F. Pereira, and R. Moore. "Semantic-head-driven generation", Computational Linguistustics, 16, 1, 30-42, 1990

[12] S. Busemann. "Best-First surface realization"; In Proceedings of the Eighth International Workshop on Natural Language Generation, 101-110, 1996.

[13] S. Russell and P. Norvig, Artificial Intelligence, A Modern Approach, 2nd Ed., Prentice Hall, 2003.

[14] G. Russell, S. Warwick, & J. Carroll. "Asymmetry in Parsing and Generating with Unification Grammars: Case Studies from ELU"; 28th Annual meeting on Association for Computational Linguistics, 205-211, 1990.

[15] R. Quirk, S. Greenbaum, G. Leech, & J. Svartvik. "A Comprehensive Grammar of the English Language". Longman, 1985.

Word Sense disambiguation for Arabic language using the variants of the Lesk algorithm

A. Zouaghi[1], **L. Merhbene**[2], **and M. Zrigui**[2]
[1]UTIC, ISI Medenine, University of Gabes, Tunisia
[2]UTIC, Faculty of sciences of Monastir, Tunisia

Abstract - *In this paper, we evaluate the variants of the Lesk algorithm to disambiguate Arabic words. In the first experiment we apply the original Lesk algorithm using the dictionary as a resource. As a second experience, we add some modifications for this algorithm, using the different similarity measures to determine the similarity relatedness between two concepts in Arabic Wordnet. We attempted to find the measure that increases the performance of the WSD. These variants affiliate a score for the most relevant sense of the ambiguous word using two different resources. .*

Keywords: Arabic Wordnet, Semantic similarities, Lesk algorithm, Window size

1 Introduction

The word sense disambiguation (WSD) allows us to find the most appropriate sense of the ambiguous word [1]. Actually, it has been about sixty years since a lot of works in this field were achieved, but they were mostly applied to English. The benefits of WSD were exploited by many NLP applications such as machine translation, information retrieval, grammatical analysis, speech processing as well as text processing [1].

In our last previous works [13] and [14], we have achieved a certain system organizing the word sense disambiguation. The problem is that the comparison between our system and the other works of Arabic is difficult, because we have used neither the same sample nor the same resources for the test. For that, in this work, we intend to evaluate a knowledge based method focusing mainly on the variants of the Lesk algorithm [11] to disambiguate Arabic words. This algorithm was used previously in many works of English WSD [4], [9] and [19]. We put into practice the original Lesk algorithm and a variant of the Lesk algorithm that uses the Arabic Wordnet (AWN). Many measures were evaluated using this modified version. Starting from these variants of the Lesk algorithm, we attempt to find the measure that increases the rate of disambiguation.

The algorithm of Lesk is used to find the gloss that matches more with the candidate glosses of the words contained in the same sentence including the word to be disambiguated. This algorithm presented some limits (cited in paragraph 4.3) to generate the correct sense. In the second part of our work, we test a modified version of the Lesk algorithm using five measures of similarities. These measures will be applied to find the similarity between each sense of the ambiguous word proposed in AWN and the senses of the other words contained in the same sentence.

This paper is structured as follows. We describe in section two the knowledge based methods of WSD and some related works. After that, in section three we describe the Arabic Wordnet. Since that in section four, we give a detailed account of the proposed experimental study for the orginal and the modified version of the Lesk algorithm. Later in section five, we present the results of that experimental study.

2 State of the art

In order to introduce the WSD research, a study of several systems was achieved in [1]. There are many approaches which are classified using the source of knowledge adapted for the differentiation of the senses. In this area, we will focus on the knowledge based methods.

They were introduced in 1970, based on the dictionary, thesaurus and lexicon. Using these resources it is possible to extract the information necessary to disambiguate words.

Some of these methods [11] tested the adequate definitions given by the electronic dictionary Collins English Dictionary (CED), Dictionary of Contemporary English (LDOCE) for the automatic treatment of the disambiguation.

Some others try to provide a basis for determining the closeness in meaning among pairs of words described by the thesaurus like Roget International Thesaurus or by the semantic lexicon like Wordnet [4].

In what follows, we present the work of Kilgarrif [8] based on the Lesk algorithm and the work of Torres and Gelbukh [17] that compared the similarity measures for the original WSD Lesk Algorithm.

In the work of Kilgarriff [8], 17 systems (supervised and non-supervised) have been tested on senseval1 (which is a company of evaluation for the WSD systems). These systems are based on the research of the correct sense for a set of previously words attributed. Every word corresponds to a task (many instances of the word to be disambiguated and two sentences containing this instance). The total number of the instances to be disambiguated was 8448. The results given by these systems were compared to different types of

performances (choosing randomly, choosing the dominant sense, Lesk algorithm). The systems that achieved the best performances were the variants of the Lesk algorithm detailed by kilgarriff and Rosenzweig [9].

They developed three variants of the Lesk algorithm. The first applies the simple Lesk algorithm that chooses for a word to be disambiguated the sense corresponding to the gloss containing the maximum number of combinations with the words of the context. The second one uses the Lesk definitions, it represents the same variants as the last one but it uses the sample of the dictionary. As far as the last variants are concerned, Kilgarriff and Rosenzweig add the corpus to the Lesk algorithm, they search for the occurrence of a word not only on the glosses of the dictionary but in the context where the sense of the word appears in the training corpus.

Some metrics were used to evaluate the WSD systems [1]. In particular, Recall is the number of correct answers provided divided by the number of correct answers to provide. Precision, on the other hand, is the number of correct answers provided divided by the total number of answers provided. The supervised systems achieved a rate of precision in order of 77%. The variants of the Lesk algorithm which was used as a baseline have overpassed the majority of the tested systems.

Torres and Gelbukh [17] used a comparison of similarity measures for WSD using the original Lesk algorithm. These similarity metrics used information drawn from Wordnet. In the experimental study they used two back-off strategies: the most frequent sense and the random sense. The Lesk algorithm shows better results when the back-off strategy focuses on random sense. Using three corpora for the tests, the accuracy of the similarity measures with different back-off strategy is between 45 and 60 percent.

For the Arabic language, we haven't found works that experiment some algorithms of WSD. However, the work of Arabic WSD applying the Lesk algorithm [18] is based on a dictionary of use and the score is calculated by counting the intersections between the sense candidate $D(s_j)$ and the words contained in the same context w (rather than their definitions).

3 Arabic Wordnet

AWN is like a dictionary in that it stores words and meanings. However it differs from the traditional ones in many ways. For instance, words in Wordnet are arranged semantically instead of alphabetically. Synonymous words are grouped together to form synonym sets, or synsets. Each such synset therefore represents a single distinct sense or concept [5].

In AWN there are four parts of speech (noun, verb, adjective and adverb), for each one there is a word sense tree that contains the synsets. The table 1 shows a comparison of AWN to English Wordnet, which details the statistics of words in each one.

From this table we can note that we are still far from having a lexical database that provides satisfactory results for the Arabic language.

Table 1. Statistics of words in AWN and English Wordnet.

		Statistics of AWN	**Statistics of English Wordnet**
Arabic Synsets	Nominal	7,961	82,115
	Verbal	2,538	13,767
	Adjectival	661	18,156
	Adverbial	110	3,621
Total		11,270	117,659
Total number of words		23,496	155,287

4 Lesk algorithm

The Lesk algorithm [11] introduced in 1986, was derived and used in several studies of Pedersen [4], Sidorov [16] and Banerjee [4]. We can also cite the work of Vasilescu [19] that evaluates variants of the Lesk approach for disambiguating words on the Senseval-2 English all words. This evaluation measures a 58% precision, using the simplified Lesk algorithm [19], and only a 42% under the original algorithm.

4.1 Original Lesk algorithm

In the first experiment we have used original version of the Lesk algorithm [11]. This algorithm exploits the similarity or relatedness between the sense definitions of the ambiguous word (MA) and the definitions of the words of its context $\{M_1, M_2, \ldots, M_i, M_n\}$. Figure 1 below provides the architecture of the Lesk algorithm, where S_A^i is the gloss (definition) corresponding to the i^{th} sense of the ambiguous word.

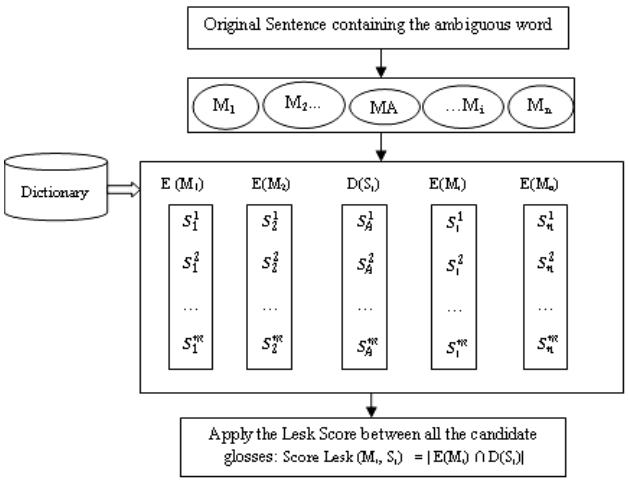


Figure 1. Architecture of the original Lesk algorithm

This algorithm uses a dictionary as a resource (see section 5.1). For every possible meaning of the word to disambiguate S_j, a definition D (S_j) is attributed. The word M (belonging to the context of the ambiguous word) is represented by the dictionary definitions E(M).

For two given words, the following score (equation 1) allows us to calculate the overlap between each possible definition of the ambiguous word and the definitions of words contained in the same sentence as the ambiguous word.

$$\text{Score Lesk (Mi, Sj)} = |E(Mi) \cap D(Sj)| \quad (1)$$

For each word to disambiguate, the algorithm assigns initially, as the best candidate, the most common sense. Another meaning is chosen if and only if its score is higher than the current best candidate.

4.2 Modified version of the Lesk Algorithm

We uses this modified version of the Lesk algorithm [4] which exploits as an input the AWN senses for the ambiguous word, as well as the words surrounding it. Then, we replace the original measure of Lesk by five measures of similarity which are intended to find the gloss that corresponds to the correct sense of the ambiguous word. A description of this modified version algorithm is detailed in what follows.

As a first step of disambiguation, we apply some pre-treatement :

- The stopwords (words that have no influence on the meaning of the ambiguous word) will be eliminated from the original sentence, using a pre-defined list (see paragraph 5.1.3).
- The words of the original sentence containing the ambiguous word are tagged manually by their synsets from Arabic wordnet. In the case when two words are represented by the same synsets, we will eliminate one of them. Besides, the words that don't contain a synset in Arabic Wordnet will also be eliminated (an example is shown in paragraph 5.2.2.1.1).
- The words that are not tagged by their synsets in AWN will be eliminated.

This first step from our work allows decreasing the number of senses to be compared with the measure of similarities.

Since that, as a second step of disambiguation the words of the original sentence will be tagged by their corresponding semantic network (tree) as shown in figure1 extracted from the AWN. In Figure 2 below, we give the hierarchy a-kind-of, extracted for the word (حسب) (hasaba). Each node represents a concept, the transcription is mentioned in bold and the translation is mentioned in italic.

This tree that represents one of the i^{th} sense of the ambiguous word (MA) حسب (hassaba), will be used to measure the similarity between the ambiguous word and the words contained in the same context {M_1, M_2, …, M_i, M_n}.

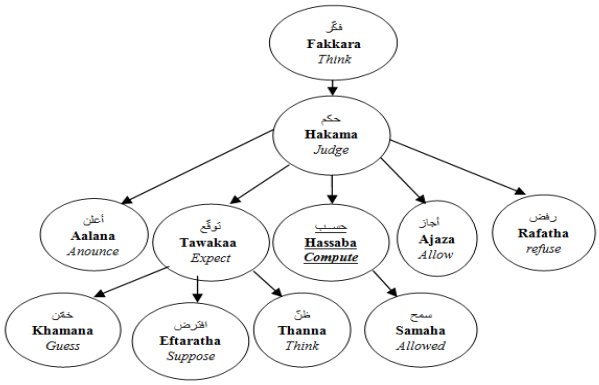


Figure 2. Tree extracted from the AWN for the word (حسب) (hasaba).

The measures cited in what follows, will be used to find the gloss that corresponds to the correct sense of the ambiguous word. These measures need the corpus and the AWN as a resource (see paragraph 5.1).

Five measures of similarity will be applied to find the gloss that corresponds to the correct sense of the ambiguous word.
The semantic similarity measure of Wu and Palmer, is based on the distance between two nodes in the hierarchy and their position relative to the root [20]. This measure is defined as follows:

$$\text{ConSim}(C_1,C_2) = \frac{2*depth(C)}{depthc(C1)+depthc(C2)} \quad (2)$$

Where Depth (C): Number of arcs that separate the root from C (the common subsequence between C1 and C2);
i: number of arc that separates C_i from the root via C;
Resnik has introduced a new factor relationship [15], called the Information Content (IC), which is defined as follows:

$$IC_{res}\ (c) = -\ log\ P(c) \quad (3)$$

Where, P(c) is the probability to find an instance of the concept C in the text.

P(c) = frequency (c) / N, where N is the total number of concepts.

The similarity measures of Resnik [15] (see equation 4), Jiang and Conrath [6] (see equation 5) and Lin [12] (see equation 6), are all linked by the factor IC which is assigned to nodes in a hierarchy, but the use of IC is a little different:

$$Sim_{res}\ (c_1, c_2) = IC\ (lcs(c_1, c_2)) \quad (4)$$

$$Rel_{jcn}\ (c_1, c2) = \frac{1}{IC(c1) + IC\ (c2) - 2*\ IC\ (lcs(c1,c2))} \quad (5)$$

$$Rel_{LIN}\ (c_1, c_2) = \frac{2*IC(lcs(c1,c2))}{IC(c1)+IC(c2)} \quad (6)$$

Where lcs represents the most specific concept that subsumes two concepts in the ontology.

Using another approach [10] assigns scores to words instead of linking to assign meaning. They define different weights to

different types of relationship in a semantic network, and use these weights to calculate score. The similarity measure Chodorow and Leacock [10] is based on the shortest distance between two nodes in the hierarchy and the maximum depth of the taxonomy. The formula is:

$$Sim_{lch}(c_1,c_2) = max\left(-log \frac{shortest\ Len(c1,c2)}{2*depth\ of\ the\ taxonomy}\right) \quad (7)$$

Figure 3 below gives more details about the modified Lesk algorithm.

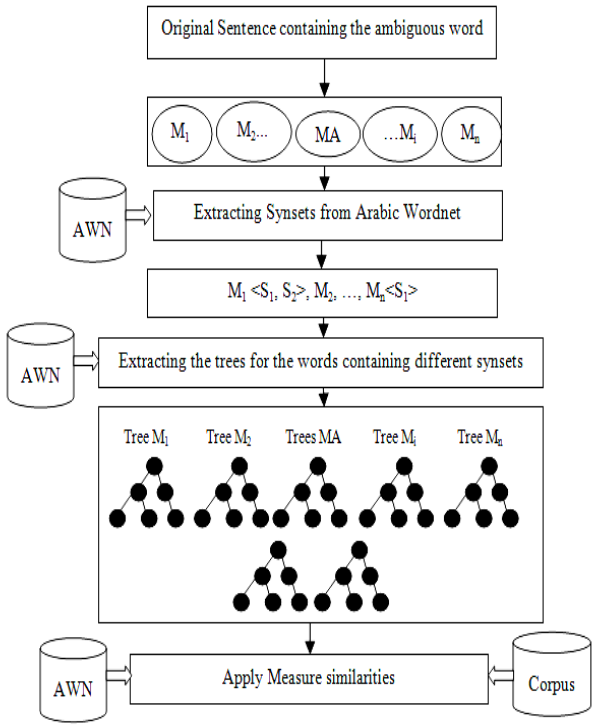


Figure 3. Architecture of the modified Lesk algorithm using AWN.

A worth noting fact is that the Lesk algorithm is very expensive because of the enormous amount of data to be compared. This algorithm is also limited to the dictionary glosses that we use, therefore the absence of a certain word can radically change the results.

Furthermore the effect of the position of the word in the sentence is not considered in the score. The closest words to the ambiguous word have generally an influence on the word to be disambiguated [21].

5 Experimental results

To check the validity of the approaches presented in the previous section, tests were conducted using some free tools. The English works were evaluated using Senseval-1 or senseval-2, however in our work we have to make our experimental data using a totally different set of resources.

5.1 Used tools and experimental data

5.1.1 Dictionnary

We use the dictionary of "Al-Mu`jam al-Wasit" [3] that contains the Arabic lexicography. Therefore, we construct a database that contains the words of an electronic version of this dictionary and their glosses. Table 2 below describes the characteristics of the dictionary.

Table 2. Description of the dictionary.

Number of letters	Number of pages	Number of gloses
29	1407	12 glosses / words

5.1.2 Latifa al-Sulaiti Corpus

We chose to work on texts dealing with multiple domains (sport, politics, religion, science, etc.). These texts are extracted from newspaper articles, which were recorded in the corpus of Latif-Al Sulaiti [2]. Table 3 below describes the characteristics of the corpus.

Table 3. Number of glosses in AWN and the dictionary.

Source	Type	Size	Average of the texts size
Public resource from the net	Written	12 glosses / words	60 sentences

These documents have the advantage of having an explicit structure that facilitates their presentation and their exploitation in different contexts to find relevant words more efficiently.

5.1.3 Stopwords

We have compiled a list of stopwords which have no influence on the semantic meaning of the sentence. This list contains 30000 words empty or stopwords. To build this list of empty words, we collected from the net pronouns, noun, names, letters, noun-verb and some words considered insignificant by humans.

Table 4. Categories of stopwords and some related samples.

	Number of stopwords	Sample of stopwords	Translation
Letter	6244	أو، ثم، من نعم، هل، قد..	or, then, from, yes, do you, May, …
Pronoun	6902	هو, نحن, أنا أنتم,…	you, is, we, I, …
Noun	5694	حَذارِ, مكانَك أمامك, سرعان,.	in front of you, soon, beware, where you are, …
Insignificant word	1272	مهما، بخلاف، على وجه الخصوص ، يوليو،...	no matter how, in contrast, in particular, july,…

5.1.4 Experimental data

Fifty words have been chosen, this number may be judged as not enough due to the problems encountered during the experimentation cited in what follows:

- The important number of glosses given by a dictionary for the ambiguous word.
- The problem of the sentence segmentation due to the ambiguity of the Arabic language [22].
- Finding the samples for the tests that can be judged as well as not so different for the process of disambiguation.
- There is a lack of words in AWN for some tested samples.

The ambiguous words have more glosses in the dictionary and the glosses of the dictionary are more extended than the glosses of AWN. Using the average of glosses in both resources (dictionary and AWN) for the 50 ambiguous words, we can measure the coverage of AWN which is 63.63%. For each one of these ambiguous words, we evaluate 20 examples per sense.

5.2 Execution of the experimentation and obtained results

Several comparative evaluations were run on a subset of the fifty ambiguous words obtained for different window sizes (three, two and one word). Thus, a context of three words would contain the three words to the left of the target word, the target word itself, and three words to the right of the target word. The samples tested in the different part of the paper under hand were used to evaluate our previous work of disambiguation [13].

In what follows we give an example, to show how to measure the similarities between two words using Arabic Wordnet. Since that, we give more details concerning the obtained results using the original and the modified version of the Lesk algorithm.

5.2.1 First experience

Let N, be the size of the context and |Sj| is the number of possible senses of the ambiguous word MA, then there is ($2 \times N \times |S_j|$) possible combinations. In the work of yarowsky [21], a study of the influence of the window size on WSD, shows that the most useful keywords for the WSD are included in a micro-context from six to eight words.

However, we have to point out that in a so large context; it's difficult to discern the key elements for determining the meaning of a word. It seems obvious that a fixed size of the context window is not adapted for all the words.

In order to solve this problem, we suggest determining the optimal size of the appropriate context for each test. We use a window size of 3 words (three words on the left and three words on the right of the ambiguous word), 2 words and one word. Figure 4 in what follows, shows as a final result of the experience the fact that the best rate of precision (P) and Recall (R) is obtained for the word (عين)(ayn), especially by using a window size of three words. The rate of precision obtained for the other window sizes doesn't help us to increase the performance of our system.

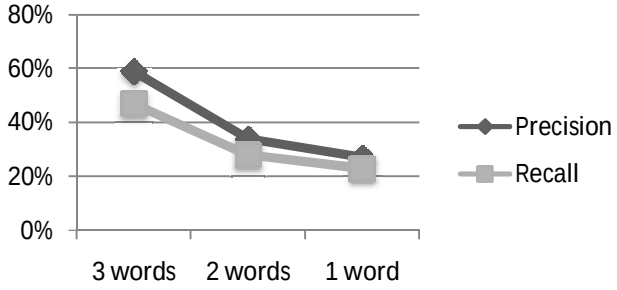


Figure 4. Results given by the original Lesk algorithm using different window size (in %).

5.2.2 Second experience

5.2.2.1 Execution of the experience

In this section we experiment the modified Lesk algorithm using the measures (explained previously in section 4.2).

5.2.2.1.1 First Step.

In the case when two words are represented by the same synsets, then we will eliminate one of them. We give a sample of a tested sentence (S) in what follows: S:

"الشّرطة تبحث عنالعميل السّرّيّالّذي كان جاسوسا لأمريكا "

[Police are searching for a Secret agent who was spying for America].

The table 6 below contains the words of S and their corresponding synsets extracted from Arabic Wordnet. The gloss of the word "جاسوسا" will be eliminated because it has had the same synset as the gloss of the word "العميل السّرّيّ.

Table 6. The corresponding synsets of the tested words.

Words	Transcription	Synsets	Translation
لشّرطة	Achortatou	شرطة، الأمن، بوليس، قوّات الشّرطة، رجال الشّرطة	Police, security, police, police forces, police
تبحث	Tabhathou	إستعلم، إستفسر، بحث	Inquire, inquire, search
العميل	Alamil	عميل سرّيّ، عين، جاسوس	Secret agent, eye, spy
جاسوسا	Jasoussan	عميل سرّيّ، عين، جاسوس	Secret agent, eye, spy
أمريكا	America		

5.2.2.1.2 Second Step

In what follows we give an example, to show how to measure the similarities between two words using Arabic Wordnet.

"ساهم الرّئيس في تقسيم المنح على المواطنين", "The president contributed to the division of grants to citizens".
First, we extract the different synsets of all the words contained in the sentence S. In this tested sentence there is no words having the same synsets, for that we go throw the second step of the experimentation.
We detail the results given by the first gloss (إعْطَاء نوع من رَمْي. رَمْي نوع من عَمَل)(Give a kind of throw. Throw a kind of work) of the ambiguous word "منح" (manaha). We calculate the number of occurrences for each word from our corpus, the distance that separate them from the ambiguous word (distance (C1, C2)), the depth of each word and the longest common subsequence (lcs).
These statistics will be used to calculate the different measures between the first gloss of the ambiguous word and the glosses proposed in AWN of the different words contained in S. The results are shown in what follows in table 7.

Table 7. Results given by the similarity measures between the different words contained in S and the ambiguous word.

Words	ConSim	Sim Res	Reljcn	RelLin	Simlch
ساهم	0,66	11	24	1.1	1,79
رئيس	0,125	8	17	0,8421	0,53
تقسيم	0 ,5	8	19	0,7619	1,23
المواطنين	0,153	8	19	0,7619	0,78

5.2.2.2 Problems encountered

As we explained previously AWN is still not achieved, and this certainly causes the problems of generating the trees of concept for some words contained in the same tested sentence.

During the experimentation of the orginal Lesk algorithm, it was assumed that for some ambiguous words none of the senses are sufficiently related to the surrounding words. In this case we give as a result the most frequent sense. This is called the back-off most frequent sense strategy used in the work of Pedersen [4].

Our experimentation is dependent on the glosses of the words in the context window; words that do not occur in any synset in AWN are ignored. The measures used in the modified version of the Lesk algorithm works only with noun-noun and verb-verb pairs. For that, we force other words in the context window to be the same part of speech as the ambiguous word.

5.2.2.3 Obtained results

Compared to the experimental results given by the original Lesk algorithm, we note that AWN increases the performance of WSD. Besides, the best similarity measure is obtained using a window size of three words. The Leacock and Chodorow measure was the best one between those proposed.

In what follows, there is a comparison between the results given by the orginal and the modified version OF THE Lesk algorithm. These results are shown below in the figure 5, for ten of the fifty words evaluated.

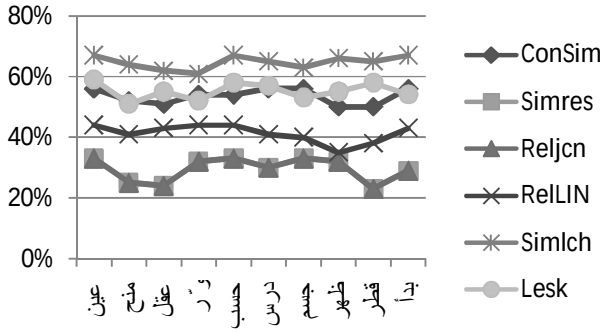


Figure 5. Comparison between the modified and the original Lesk algorithm

During the experimental process, we try to make sample of sentences that contains the words belonging to the AWN.

5.2.3 Comparison with our previous work

The words evaluated with the variants of the Lesk algorithm and the test samples, are used in the experimental study of our algorithm WSD-AL [13]. For that in the table 8 below, we compare the results obtained previously with the various results given by this experimental study.

Table 8. Comparison of the results obtained by the algorithm WSD-AL and the variants of the Lesk algorithm.

Methods		Precision
WSD-AL		78 %
Original Lesk Algorithm		59 %
Modified Lesk Algorithm Used the AWN	Wu and Palmer (ComSim)	56 %
	Resnik (Sim_{res})	33 %
	Jiang and Conrath (Rel_{jcn})	33 %
	Lin(Rel_{LIN})	44 %
	Leacock and Chodorow (Sim_{lch})	67 %

As we see in the table 8, the WSD-AL system achieves the best performance in terms of precision. In this comparison, we apply the same experimental data that we have used before in the experimental study of the WSD-AL system.
We note that the WSD-AL algorithm [13] matches the original sentence with the context of use that corresponds to the gloss of the ambiguous word. After that we choose the most relevant score obtained corresponding to the exact sense.

6 CONCLUSION

In this paper we evaluate the variants of the Lesk algorithm and the use of the AWN to perform WSD in Arabic. These

variants affiliate a score for the most relevant sense of the ambiguous word using two different resources. For a sample of fifty ambiguous Arabic words that are chosen by their number of senses out of contexts (the most ambiguous words), the modified version of the Lesk algorithm achieves a precision of 67%.

The best rate of precision is obtained for a window size of three words using the original Lesk algorithm and two words using the modified Lesk algorithm that uses AWN. For the modified version of the Lesk algorithm, between those proposed measures, the Leacock and Chodorow measure achieves the best performances of disambiguation. We propose that in future works; we ameliorate the correspondence between words and their glosses, to automate the step of tagging the words with their corresponding synsets from AWN.

7 References

[1] Agirre, E. and Edmond,P."Word Sense Disambiguation: Algorithms and Applications." Springer, (www.wsdbook.org), 2006.

[2] Al-Sulaiti L, Atwell, E. "The design of a corpus of contemporary Arabic." International Journal of Corpus Linguistics, vol. 11, pp. 135-171, 2006.

[3] Ben Mukarram M., Al-ifriqi, AL-Misri, Ibn Manzour. Édition libanaise imprimée du Lisàn al-'arab -- Ibn Manzûr, "Lisàn al-'arab," 15 volumes. Beyrouth, 1290.

[4] Banerjee, S., Pedersen, T. "Adapting the Lesk Algorithm for Word Sense Disambiguation to WordNet." Submitted in partial fulfillment of the requirements for the degree of Master of Science. 2002.

[5] Elkateb, S., Black, W. J. "A Bilingual Dictionary with Enriched Lexical Information." Proceedings of NEMLAR, Egypt, Arabic Language Tools and Resources: 79-84, 2004.

[6] Jiang, J., Conrath, D. "Semantic similarity based on corpus statistics and lexical taxonomy." In Proceedings on International Conference on Research in Computational Linguistics, Taiwan, 1997.

[7] Hirst, G., St Onge, D. "Lexical chains as representations of context for the detection and correction of malapropisms." In Christiane Fellbaum (editor), WordNet: An electronic lexical database, Cambridge, MA: The MIT Press,1998.

[8] Kilgarriff., A. "SENSEVAL: An Exercise in Evaluating Word Sense Disambiguation" Programs In Proc. LREC, Granada, pp. 581-588, 1998.

[9] Kilgarriff, A., Rosenzweig, J. "English SENSEVAL: Report and Results." In Proceedings of the 2nd International Conference on Language Resources and Evaluation, LREC, Athens, Greece, 2002.

[10] Leacock, Chodorow, M. "Combining local context and WordNet sense similarity for word sense identification." In WordNet, An Electronic Lexical Database. The MIT Press, 1998.

[11] Lesk, M. "Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an .ice cream cone" In proceeding of ACM Special Interest Group for Design of Communication Proceedings of th 5the annual international conference on Systems documentation, p. 24 – 26, ISBN 0897912241, 1986.

[12] Lin, D. "An information-theoretic definition of similarity." In Proceedings of 15th International Conference On Machine Learning , 1998

[13] Merhbene, L., Zouaghi, A., Zrigui, M. "Ambiguous Arabic Words Disambiguation." SNPD: 157-164, London, England, 2010.

[14] Merhbene, L., Zouaghi, A., and Zrigui, M. "Arabic Word Sense Disambiguation". ICAART 2010, Valencia, Spain, 2010.

[15] Resnik, P. "Using information content to evaluate semantic similarity in a taxonomy." In Proceedings of the 14th International Joint Conference on Artificial Intelligence, Montreal, 1995.

[16] Sidorov, G., Gelbukh, A. "Word Sense Disambiguation in a Spanish Explanatory Dictionary."Proceedings TALN-2001, pp. 398-402, Tours, France, 2001.

[17] Torres, S., Gelbukh., A.. "Comparing Similarity Measures for Original WSD Lesk Algorithm." Alejandro Buchmann (Ed.). Advances in Computer Science and Applications. Special issue of J. Research in Computing Science, ISSN 1870-4069, N 43, pp.155–166, 2009.

[18] Tlili-Guiassa, Y., Merouani, H. "Désambiguïsation sémantique d'un texte arabe. " Conference TALN'06 , 2006.

[19] Vasilescu, F., Langlais, P., Lapalme, J. "Evaluating Variants of the Lesk Approach for Disambiguating Words." LREC, Portugal, 2004.

[20] Wu, Z., Palmer, M. "Verb Semantics and Lexical Selection." Proceedings of the 32nd Annual Meetings of the Associations for Computational Linguistics, p.133-138, 1994.

[21] Yarowsky D. "One sense per collocation." In Proceedings of the ARPA Workshop on Human Language Technology, Princeton, pp. 266-7, 1993.

[22] Zouaghi A., Zrigui M., Antoniadis G. "Compréhension automatique de la parole arabe spontanée : Une modélisation numérique". Revue TAL Varia. No. 1, Janvier 2008, Vol. 49, 2008.

Arabic Call system based on pedagogically indexed text

Mohamed Achraf Ben Mohamed [2,3], **Dhaou El Ghoul**[1], **Mohamed Amine Nahdi** [1], **Mourad Mars**[1,2]
Mounir Zrigui[2,3]
[1]LIDILEM, University of Stendhal, Grenoble3, France
[2]UTIC Laboratory, University of Monastir, Tunisia
[3]Faculty of sciences of Monastir, Tunisia

Abstract - *This article introduces the benefits of using computer as a tool for foreign language teaching and learning. It describes the effect of using Natural Language Processing (NLP) tools for learning Arabic. The technique explored in this particular case is the employment of pedagogically indexed corpora. This text-based method provides the teacher the advantage of building activities based on texts adapted to a particular pedagogical situation. This paper also presents ARAC: a Platform dedicated to language educators allowing them to create activities within their own pedagogical area of interest.*

Keywords: ICALL, Pedagogical Text Indexation, Natural Language Processing, Language Learning.

1 Introduction

Computer-assisted language learning (CALL) software emerged since the 1960's and have been making continuous progress, especially in the recent years with the vulgarization of new technologies and the development of new teaching methods. There are three distinct phases in the evolution of CALL [14]:

- Communicative CALL (1970's-1980's) was based on the behaviorist theories of learning and teaching, this approach is based on drills and repetition.
- Structural CALL (1980's-1990's) rejected behaviorist approach and insisted that CALL should focus more on using forms rather than on the forms themselves.
- Integrative CALL (21st century) is based on communicative CALL with the integration of two advanced technologies into language teaching: multimedia, on one hand and the development of networks and the Internet on the other.

2 Motivations behind CALL emergence

Since computers became widespread, education became more concrete and more assimilated by learners who are no longer just passive listeners, especially in language learning, rather as active participants [5, 10, 13]. In the following we are going to detail the advantages of CALL and we will justify the huge potential it has acquired.

2.1 Learner autonomy

Teaching based on new technologies like the Internet, can help learners improve their language skills by helping them develop strategies for self-learning and promote self-confidence. Student motivation is increased, especially when a variety of activities is proposed, which makes them feel more independent [1].

2.2 Authentic learning

The computer can improve the authenticity of the language enabling learners to see and hear native speakers interacting in the language, not only verbal but also devoted wholly or partially in the facial expressions and gestures [14]. The Internet also facilitates the use of a domain specific language [12]. It is an endless source of authentic materials. Learners will have the opportunity to use various resources of authentic materials which incorporate graphics, audio and video, students can improve their ability to speak, to increase their comprehension skills, expand their vocabularies, to test their pronunciation, and practice their reading and writing.

2.3 Personalized learning

Timid or fearful learners can be greatly beneficiaries of this individualized learning centered on learner. 'Fast' learners can also reach their full potential without preventing their peers to work at their own rhythm. In language learning, teachers are confronted by several skills and learning capacities [7]. Computers can help teachers to 'orchestrate' different rhythms of different learners by giving them adaptable learning methods and tools. This encourages learners to take more responsibility for their own learning.

2.4 Benefits of using multimedia

The use of computers in language teaching has come to break with the old methods of learning that were passive and boring. Computers have helped to integrate photos and video. Previously considered abstract concepts become, through simulation for example, more real and understandable and learners can study more actively [4, 9].

2.5 Learning beyond the limits of time and space

Through computer networks, learners are now able to access knowledge wherever and whenever they want, not only that, particularly the web has democratized learning by making it accessible to everyone regardless of their ethnicity, origin or income [12]. In addition the web offers the opportunity to create learning groups including learners and teachers.

3 Pedagogical text indexation

3.1 Definition

«Pedagogical indexation is an indexation performed according to a documentary language that allows the user to search for objects to use in teaching. » [7].

Our center of interest is language learning and our aim is mainly educational, in another way the expectation of the user (teacher) of a document indexed for learning would be to extract objects that can be used as part of its course (Arabic learning in our case) with an application of CALL, this implies that this base of indexed texts must ensure, through its interface, the following features [6, 8]:

- Enabling teachers to make requests based on criteria consistent with the teaching of languages.
- Allow teachers to add their own texts to the base.
- Language teachers are not necessarily very knowledgeable in computers; the interface should be as friendly as possible.

3.2 Educational Metadata Standards

Educational Metadata Standards are initiatives in the field of standards for e-Learning and information and communication technology (ICT) designed to describe objects by metadata [9]. The term metadata is used to define all the technical and descriptive information added to documents to better describe them. Metadata were originally proposed to improve Web search, they are to enrich the documents published by a set of information to better identify them: name of the author and publisher, publication date, Summary of content, etc...

The oldest standard concerned with the description of learning objects is the Dublin Core Metadata Initiative (DCMI), there are other initiatives resulting directly from DCMI, Getaway to Educational Material (GEM) and Educational Network of Australia (EdNA), however, the most common standard used today is Learning Object Metadata (LOM) [9]. For the description of texts constituting the corpora of our platform ARAC, we pick up the LOM standard.

4 ARAC Platform

4.1 Presentation

Our modules-based platform ARAC (ARAbic Call) has been developed to allow users to manage and exploit a corpus from a single interface with tools both powerful and easy to use to automate the work needed. The platform ARAC was developed in an ASP mode (Application Service Provider); this mode avoids the technical constraints and allows users to automatically benefit from any updates. Generally, only a password and login are required to access the platform at any time (Authentication). This technology (ASP) requires no special configuration and therefore excludes any costs of installation, operation or maintenance engendered by the most heavy software solutions. Throughout the development of this platform we have tried to make the technical side transparent from the user (teacher) who should only focus on the educational aspect.

4.2 Architecture

ARAC platform consists of four modules with high synergy; it was developed in accordance with the general functional diagram of learning system [2, 3, 11] (Figure 1).

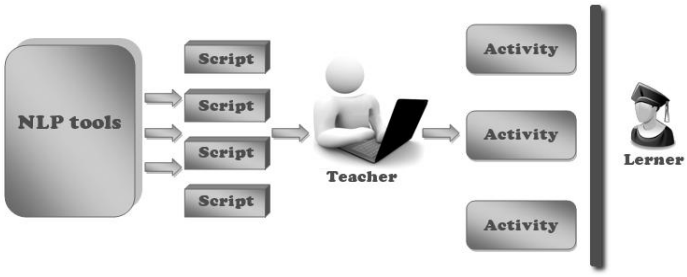


Figure 1 - General functional diagram of learning system

Figure 2 presents the architecture of our platform; in the following we will give an overview of each module.

4.2.1 Text management module

This is the input module of the platform by texts encoded using the standard UTF-8. The choice of this coding was justified by the fact that, firstly, most of the Arab digital textual resources are encoded using this standard, and secondly because the standard UTF-8 is supported by the most popular browsers. [18] The management module corpus is responsible from two tasks, namely: classification and annotation.

- Classification: The corpus is composed of several themes created by the administrator and each containing texts. Very often the teacher is looking for a text in a particular subject.

- Annotation: ARAC platform offers two forms of annotation, the first is automatic (figure 3), it allows the user to identify particles such as particles of coordination or interrogative particles, the second is manual, it allows the user to identify any other lexemes [17] [19].

The algorithm of automatic annotation (Annotatik) is shown in the table 1.

Table 1- The Pseudo code for automatic annotation algorithm.

Algorithm Annotatik (F, T)

Input: File F , T // F:Corpora, T:taxonomies
Output: Nothing (but insert in the database)

```
Lock file(F) // Prevent concurrent  file access.
lexemes[] ← split file content (F) //retrieve the lexemes in a table.
taxonomies[]← split file content (T) //retrieve the taxonomies in a table.
counter ← 0
rasult[] = array
for(j=0; j<count(taxonomies); ++j){
   for(i=0; i<count(lexemes); ++i){
      if(taxonomies[j] == lexemes[i] ){ // looks for elements of the taxonomies table present in the text.
                result [counter ++] = i; // group results.
      }
   }
}

if( count(result) > 0) { // at least one element has been found.
   for(k=0; k<count(result); ++k)
      insert into the database (result[k])
```

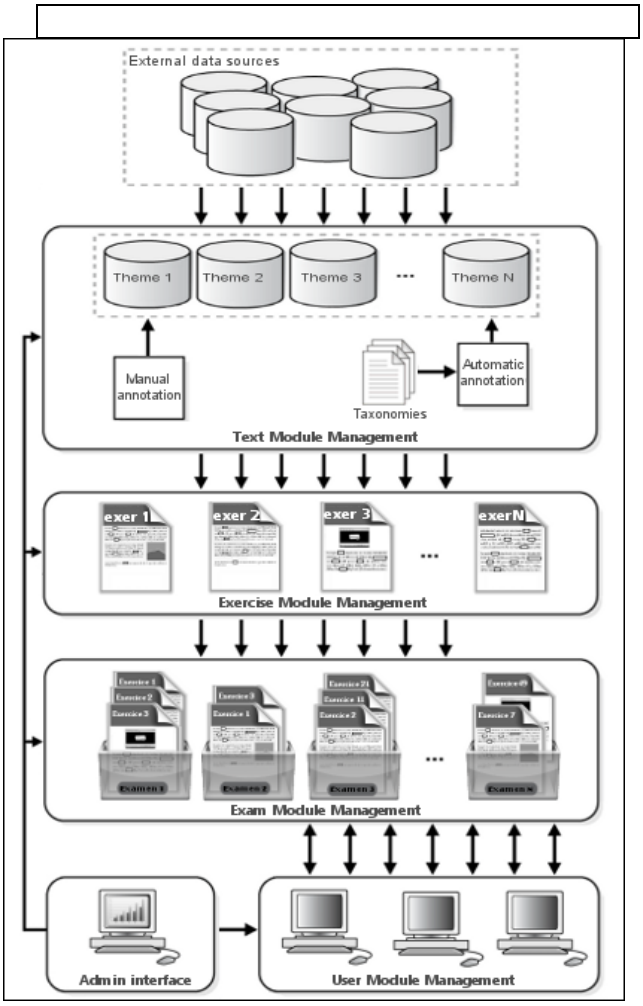


Figure 2 - Architecture of the Platform ARAC

4.2.2 Exercise management

With this module, we move from the acquisition of text step to the exploitation of text step. The production of an exercise requires two steps: first, the choice of text responding to the request of the teacher, and second, formatting the exercise using a WYSIWYG editor (figure 4).

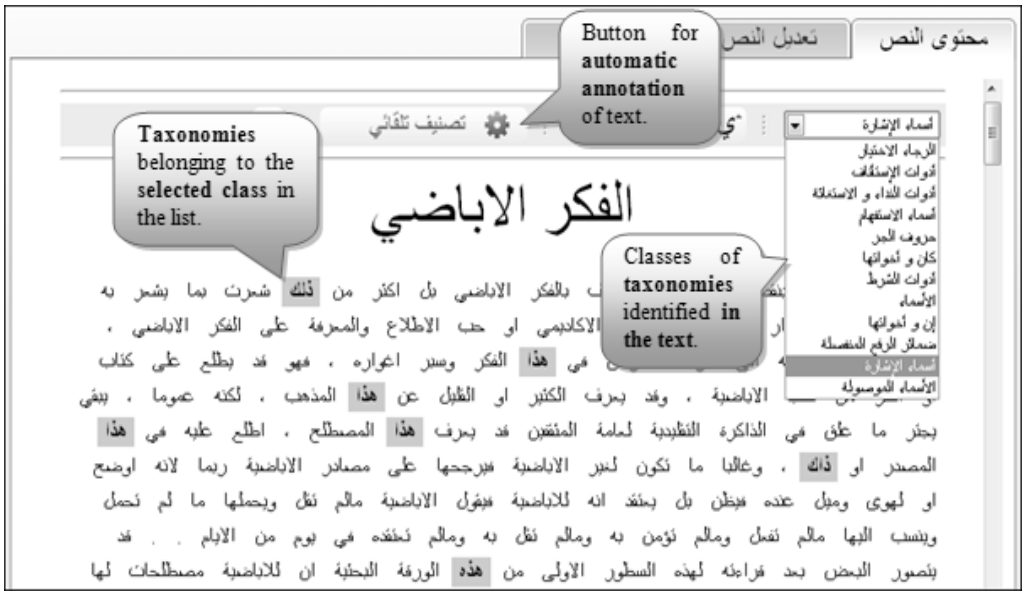


Figure 3 - Automatic annotation

Figure 4 - Exercise formatting.

4.2.3 Exam management

An exam is an ordered selection of exercises selected from the available exercises. After being created, the exam will be assigned to students by the teacher. Figure 5 shows the final overview of the activity.

Figure 5 - Preview of a non accomplished exam

4.2.4 User Management

User management covers several aspects such as user account creation, exams monitoring (accomplished or not), the performances (Figure 7) and deleting accounts.

Performance is given by the following formula:

$$Performance = \frac{Number\ of\ Correct\ Answers}{Number\ of\ Questions} \times 100 \quad (1)$$

4.3 Example of activity

For each activity, the learner will be asked to complete the empty boxes with appropriate terms, and once the form is validated, the student will receive a page containing the answers to the exam and the results are verified (Figure 6). This correction page will indicate, by a set of colors, where the learner has done good (green) or bad (red) responses. In addition to that, the correction will be provided with statistics.

Figure 6 - Preview of an accomplished exam.

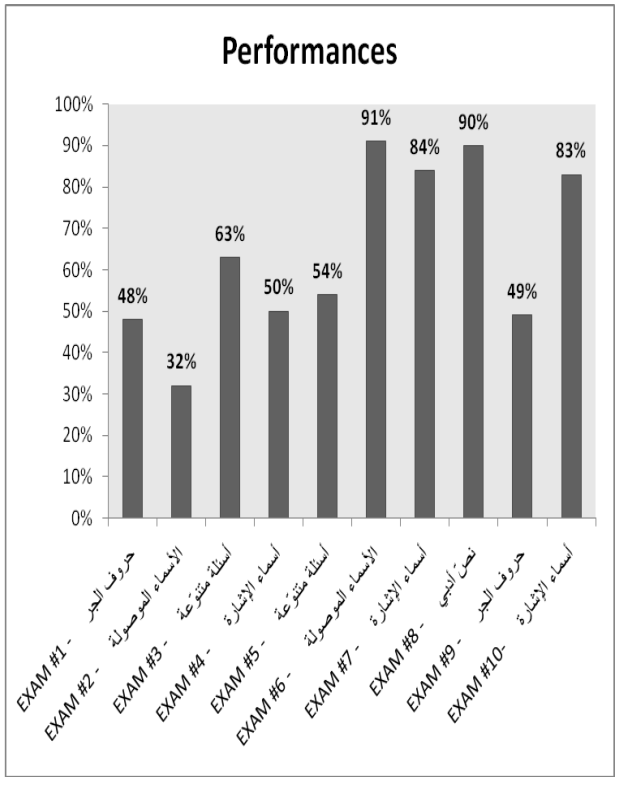


Figure 7 - Performances achieved by a student

5 Conclusions

We developed a tool [16] for creating and exploiting a database of text indexed pedagogically; this tool will allow language teachers to create activities within their own pedagogical area of interest. We chose to develop the ARAC platform in ASP mode, thus making the implementation and updating of this tool as simple as possible and also to ensure full mobility of data access whether to the teacher or learner, independently of the machine and operating system in use (platform independent).

For future work we plan to improve our platform ARAC in some aspects, such as the possibility of treating any type of data (e.g. RSS), we can also consider automating the classification, we also plan to add an analysis and feedback module.

6 References

[1] Chanier T. "Acquisition des langues assistée par ordinateur (ALAO). Habilitation à diriger des recherches". Blaise Pascal University - Clermond-Ferrand II, 1995.

[2] Georges ANTONIADIS, Sandra ECHINARD, Olivier KRAIF, Thomas LEBARBÉ, Claude PONTON. "Modélisation de l'intégration de ressources TAL pour l'apprentissage des langues : la plateforme MIRTO". alsic.org ou alsic.u-strasbg.fr, Volume 8, 2005, pp. 65-79.

[3] Georges Antoniadis & Claude Ponton. "MIRTO: un système au service de l'enseignement des langues". UNTELE 2004, 17-20 mars 2004, Compiègne.

[4] Georges Antoniadis, Sandra Échinard, Olivier Kraif, Thomas Lebarbé, Mathieu Loiseau et Claude Ponton."NLP-based scripting for CALL activities". LIDILEM, Université Stendhal Grenoble 3, COLING Workshop - Aout 2004.

[5] Loiseau M. "La description de ressources pedagogiques: etat de l'art et application aux ressources textuelles pour l'enseignement des langues". Journée d'étude de l'ATALA délocalisée à Grenoble, Traitement Automatique des Langues et Apprentissage des Langues.Grenoble 2004.

[6] Loiseau M. "Vers la création d'une base de données de ressources textuelles indexée pédagogiquement pour l'enseignement des langues". Mémoire de DEA en Sciences du Langage, Stendhal University, Grenoble, France, 2003.

[7] Loiseau M. "Élaboration d'un modèle pour une base de textes indexée pédagogiquement pour l'enseignement des langues", Ph. D. thesis: Stendhal University, Grenoble 3: 2009.

[8] Loiseau M., Antoniadis G., Ponton C. "Pedagogical text indexation and exploitation for language teaching". 3rd International Conference on Multimedia and ICTs in Education. juin 2005, Cáceres, Extremadura (Espagne).

[9] Mathieu Loiseau, Georges Antoniadis, Claude Ponton: "Model for Pedagogical Indexation of Texts for Language Teaching". ICSOFT (ISDM/ABF) 2008: 212-217

[10] Mohsen Maraoui, Georges Antoniadis, Mounir Zrigui: "CALL System for Arabic Based on Natural Language Processing Tools". IICAI 2009: 2249-2258.

[11] Mohsen Maraoui, Georges Antoniadis and Mounir Zrigui: "SALA: Call System for Arabic Based on NLP Tools". IC-AI 2009: 168-172

[12] Mounir Zrigui, "Contribution au traitement automatique de l'Arabe. HDR en informatique", Stendhal University, Grenoble 3, France, 2008.

[13] Mourad MARS, Georges Antoniadis, Mounir Zrigui: "Nouvelles ressources et nouvelles pratiques pédagogiques avec les outils TAL", ISDM32, N°571, April (2008).

[14] Warschauer, M. & Healey, D. "Computers and language learning: An overview". Language Teaching, 1998, 31: 57-71.

[15] Mourad Mars, Georges Antoniadis, Mounir Zrigui: Statistical Part Of Speech Tagger For Arabic Language. IC-AI 2010: 894-899

[16] Anis Zouaghi, *Mounir Zrigui*, Georges Antoniadis: Compréhension automatique de la parole arabe spontanée; Revue Tal, volume 49- n 1/2008.

[17] Rami Ayadi, Mohsen Maraoui, Mounir Zrigui: Intertextual distance for Arabic texts classification. ICITST 2009: 1-6

[18] Mounir Zrigui, Mbarki Chahad, Anis Zouaghi, and Mohsen Maraoui: A Framework of Indexation and Document Video Retrieval based of the Conceptual Graphs. CIT 18(3): (2010)

[19] Rami Ayadi, Mohsen Maraoui, Mounir Zrigui: SCAT: a system of classification for Arabic texts; International Journal of Internet Technology and Secured Transactions 2011 - Vol. 3, No.1 pp. 63 - 80

@rabLearn: a Model of NLP Tools Integration in ICALL Systems

Mourad Mars[1], Georges Antoniadis[1], Mounir Zrigui[2]

[1] *LIDILEM Laboratory Grenoble - France*
Mourad.mars@e.u-grenoble3.fr
Georges.antoniadis@u-grenoble3.fr
[2]*UTIC Laboratory Monastir - Tunisia*
Mounir.zrigui@fsm.rnu.tn

Abstract **— This paper describes the development of an Intelligent Computer Assisted Language Learning (ICALL) platform for Arabic using Natural Language Processing (NLP) tools, this environment called @rabLearn. As a first step, we have developed a Part Of Speech Tagger (POS tagger) for Arabic. In a second step, we used this NLP tool to create educational activities for Arabic learners by the use of @rabLearn (an NLP-based authoring system). We will focus, in this paper, on the integration of an NLP tool like our morphological analyser in a platform of language learning to generate activities intended for learner. We prove that this tool can make platforms more powerful by creating various kinds of activities.**

Keywords **— ICALL, Natural Language Processing, Arabic Language Learning, POS tagge, NLP tools.**

I. INTRODUCTION

The introduction of the TIC (Technology of Information and Communication) into languages learning platforms seems subjected to the need for close monitoring of technological advances. Although a transformation and an improvement of the output always levelled, the meticulous observation of the multiple devices and software which we carried revealed faults, which only the use of the NLP is able to correct it [9, 10, 12]. In this context, our approach can be expressed by: how could systems of ICALL benefit from procedures, solutions and tools of natural language processing? [16, 19]. Some platforms dedicated to languages learning, like @rabLearn for Arabic language, provide answers to this question. @rabLearn allows the creation of educational activities and it facilitates the conception and enhances transparency to teachers of Arabic language to conceive several activities using NLP tools; this process is made with simple and very ergonomic graphical interfaces adapted to all teachers even those without any knowledge.

This approach is intended for qualified teachers of language or teachers having only little or no knowledge in NLP or in computer sciences. The technical side is transparent for the user and only the didactic and pedagogic side is visible and available.

In this paper, we will explain how we can integrate an NLP tool like our morphological analyser in a platform of language learning to generate activities intended for learner. We prove that this tool can make platforms more powerful by creating various kinds of activities.

In the field of the interaction teacher-learner and ICALL in general, the feedback holds particular place. Indeed, it is considered as one of the essential factors of language learning. We will study the aspects and kinds of feedback specific ICALL, and we will seek to know in what ways the analysis of the learner's answer by the computer improve the quality of the feedback. [15, 17, 18].

Before presenting our platforms and the used techniques, let's describe the structure of the paper.

The rest of this paper is structured as follows. First, related works on Arabic language learning programs are listed. We also discuss limitations of some Arabic language learning systems, especially platforms without any NLP tool. After that, we describe @rabLearn: our proposed Arabic ICALL system based on NLP tools. We start by giving the general architecture; we introduce our POS tagger (NLP tool) and we continue, in the following sections, the presentation of the main components of our Arabic ICALL system (interfaces, activities, feedback, and learner's results). Finally, we conclude the paper and give directions for future work.

II. CURRENT LANGUAGE LEARNING SYSTEMS: A BRIEF OVERVIEW

In this section, we present a brief overview of different researches in Arabic CALL recently achieved.

The language Studies Department at London Guildhall University has developed a web-based CALL system consisting of an authoring program that permits teachers to take their own text and

convert it into any of 10 different varieties of exercises such as gap filling, multiple choice, jumbled text, etc., and which all runs in a browser. The system is asked to produce course material for university's Arabic classes.

Arabic Visual Interactive Syntax Learning (ArabVISL) makes possible for learners to analyse a (still limited) number of Arabic sentences and to gain access definitions and explorations of Arabic grammatical terms. ArabVISL is, however, based on non-automatic parsing – that is, all sentences are manually tagged and analysed – and this limits the scope of what ArabVISL can offer in terms of gramma drills, exercises, etc.

Shaalan (2003) developed an Arabic grammar checker, called Arabic GramCheck. Arabic GramCheck looks for common Arabic grammatical problems, describes the problem for you, and offers suggestions for improvement. This program is useful in pointing to problems believed typical of native speaker writing. Thus, she/he can avoid such problems in the future.

There is other CALL software on the Internet, designed using authoring systems such as (Course builder, Hot Potatoes or Netquizz). Such activities poses several problems as the rigidity of software (the data used are predetermined and can not be altered or enhanced) and the not adaptability of course to the language skills of learners. The only advantage of such exercises contribute to those written on paper is the interactivity of the computer.

The use of the NLP for the design of ICALL software is currently the best solution to solve these problems.

III. @RABLEARN: ICALL SYSTEM BASED ON NLP TOOLS

A. Architecture of @rabLearn

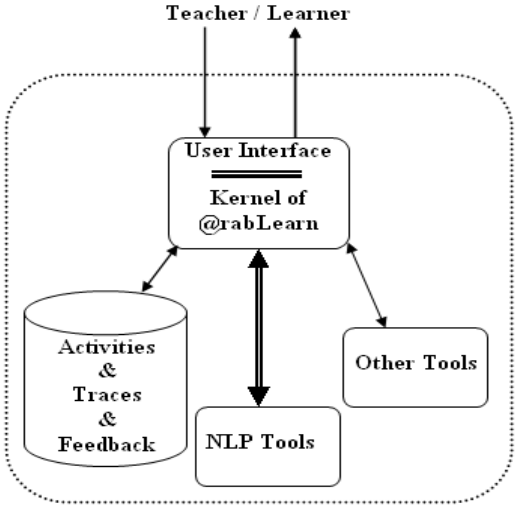


Fig. 1 @rabLearn: General architecture

This figure shows the overall architecture of the proposed Arabic ICALL system. This system consists of the following components: The user interface provides the means of communications between the teacher, the learner and the Arabic ICALL system. The data base includes activities generator, NLP tools, educational activities, communication tools, feedback's procedures, and an intelligent evaluation component.

B. POS Tagger as NLP tool

Arabic is one of the Semitic languages; known for its rich and systematic but very complex morphological structure and also for its several problems in natural language processing (agglutinative form, run-on word, free concatenation and orthographic variation).

What is the best sequence of tags which corresponds to a given sequence of words? The objective of POS tagging is to resolve ambiguities, the use of a POS tagger can solve this problem; it attempts to assign the correct tag or lexical category to all words in a text.

This is an example of tagged sentences (POS tagger output):

(1) ktb/VBP rsAlp/NNS ./PUNCT

(2) ktb/NN jmylp/Adj ./PUNCT

In both sentences, automatically assigning one tag per word is not easy.

We integrate this NLP tool in our environment to make easy and automatic the task of activities generation.

The Pos tagger represents the base of NLP applications, especially in the field of languages learning. In this kind of applications, the output mustn't be wrong: the application cannot give to its user an approximate result.

C. Interface of @rabLearn

The Computer allows access to all NLP tools, language resources and the internal view scripts for administrator (computer science specialist). It is useful for updating the tools, resources and scripts.

The language teacher interface is simple and easy to control (no computer knowledge is required), which provides access to the external view of scripts to create activities, and. The parameters of each activity should be kept simple (too many parameters could discourage the user), and friendly (important information should be readily apprehensible), and user is not supposed to master a complex formalism, the definition of parameters should be intuitive.

D. Activities generation

From a purely educational level, the design of activities is a task carried out by teachers of languages, through a specific interface. An activity

corresponds to what is traditionally designated as an exercise given to the learner to enable him to reach a goal. Generating activities is the implementation of a didactic script. An activity is a didactic purpose which may be used by learners, in an autonomous manner.

A large variety of activities are proposed in @rabLearn such as: Linguistic analysis of words between brackets or a sentence, Verb conjugation, words according to certain morphological feature.

- Steps of activities generations (fill in the blanks):

The first step concerns the choice of activity to implement from a list of available activities. Once the activity is chosen, teachers must choose from the database (texts shared by other teachers) or upload the text for its activity. Next, the POS tagger is used to attribute for each word from the text, the right grammatical category. Teachers must verify if all categories are correct, according to parameters available in simplified form, through a panel. The following task is to specify parameters for the activity (the person, time, gender, number, mode of conjugation…), then the teachers associate to this activity the right feedback and the activity can be proposed to learners.

Fig.2 Activity Interface

E. *Feedback of the teacher and feedback of the computer: brief comparison*

Feedback aiming at the correction of learner's errors presents many important differences according to whether the learner is lavished by the teacher in classroom or by a language learning environment [13, 14]. A good amount of these differences were presented as advantages of the teaching machine. The following table contrasts these positive factors of the feedback in situation of CALL with the weaknesses of error treatment practices by teachers in classroom situation. [5,6]

TABLE I

COMPARISON OF THE FEEDBACK IN ICALL AND IN CLASSROOM SITUATION

ICALL Feedback	**Classroom Feedback**
Applies to each incorrect response.	Teacher correct error in one part of the lesson and forgot it in another.
Private	Public
Instantaneous	Instantaneous in oral interaction, delayed when written work is corrected.

F. *Types of activities and feedback*

In order to establish a classification of the principal types of existing activities and feedback in @rabLearn, we propose this table to classify the types of asked questions by the ICALL system, which condition the possible types of answers for learning and - at the third time of the interaction - the feedback itself [14].

TABLE II

PRINCIPAL TYPES OF ACTIVITIES AND FEEDBACK IN @RABLEARN

Level	**Quest-ion**	**Answer And New answer**	**Feedback**
1	Text-hole	True or False - Guidance towards a new choice	Correct answer: Positif feedback Incorrect answer: Negative feedback
2	True or False	True or False - Only one possibility	Correct answer: Positif feedback Incorrect answer: Negative feedback
3	QCM	Choice of an answer among Other - Many possibility (limited in number)	Correct answer: Positif feedback Incorrect answer: Negative feedback
4	Lacunar exercise	Answer limited to one or more words - Guidance towards a new choice (in a restricted unit but not limited)	Correct answer: Positif feedback Incorrect answer: Negative feedback
…			

G. The Feedback architecture and mechanism

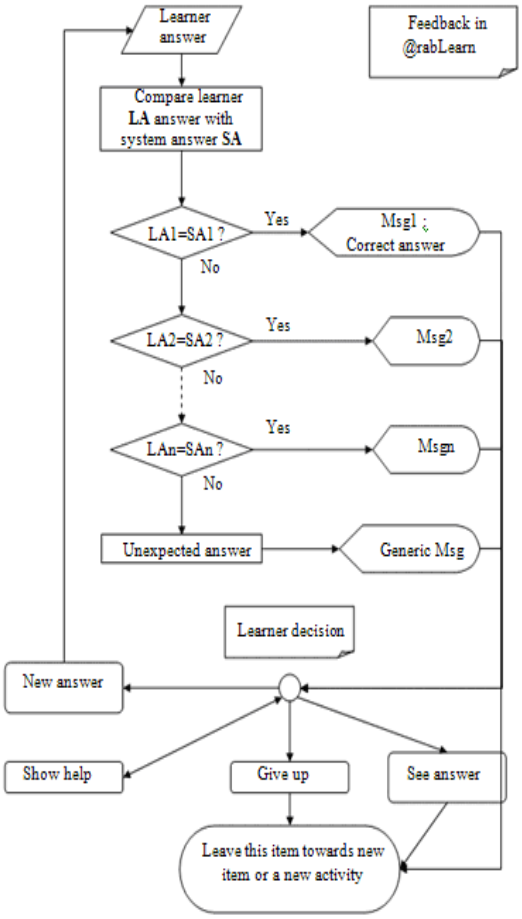


Fig.3 Diagram of answer analysis and interaction in @rabLearn

This figure show the mechanism of answer analysis in @rabLearn, our approach is based on helping learner to find the right answer with the use of an interactive feedback, starting by giving some indicator to guide learner. Our intervention depends on activities and user answer [13]. We called this kind of dialog 'cooperation sequence'.

H. Results of activities and traces

The recording traces of the actions on which the learner can make a synthesis of system use. The traces are organized by session work and record any action done by the learner by date which allows calculating the time spent in each response in all activities and a saw to the teacher about how is used system and assess whether an activity should be modified.

The teacher has an interactive interface which allows him to see many statistics about learners.

Fig.4 Teacher's statistics interface

IV. Conclusions

The aim of our work was to design and implement an ICALL system for Arabic language, based on NLP tools. In conclusion, the ordinary NLP tools can be diverted and handled to propose a reflexive approach in languages learning and linguistics. However, in spite of its obvious interest, NLP does not adapt in an immediate way to ICALL systems. The major difficulty is related to the imperfect performances of the NLP tools which must be used with caution in the learning applications. Indeed, if some simple applications (as morphological analysis) give good performances, they are never free from error.

We plan to enrich the present system, e.g. make the system available on the Internet to serve remote learners worldwide (especially learners of Arabic as a second language), and extend the grammar coverage to include more advanced grammar levels.

Other important point is: The dimension relates to communication and interaction between the tutor and the learner. Especially important in this interaction is the aspect of a bi-directional interaction: The learner does not only want to have a tutorial feedback but wants to give feedback to the tutor as well. This kind of interaction must be integrated in our next version of @rabLearn.

Acknowledgment

The morphological analyser used, as an NLP tool, in this project is part of three large research projects, the first one titled "Oreillodule" which is a tool for automatic speech recognition, translation, and synthesis for Arabic language. The others are "MIRTO" & "@rabLearn" which presented in this paper: two Intelligent Computer Assisted Language Learning (ICALL) platforms. The integration of our POS tagger in both ICALL systems will help teachers to generate, automatically, pedagogical activities and enjoy a powerful feedback.

REFERENCES

[1] Allison Piguet, Daniel Peraya. Creating Web-integrated learning environment: An analysis of WebCT Authoring Tools in Respect to Usability. Australia Journal of Educational Technology, Vol. 16, No. 3, pp. 302-314. 2000

[2] Antonija Mitrovic, Brent Martin. Evaluating Effectiveness of Feedback in SQL-Tutor. In the Proceedings of International Workshop on Advanced Learning Technologies, IWALT 2000, pp. 143-144, Palmerston North. 2000

[3] Dominique Hémard & Steve Cushion. Designing a CALL Package for Arabic While Learning the Language Ab Initio, Computer Assisted Language Learning (CALL): An International Journal, Vol. 16(2-3): 259-266, SWETS & ZEITLINER publisher. 2003

[4] Everhard Ditters, Nelleke Oostdijk, and Keith Cameron. Processing Arabic: Computer Assisted Language Learning, Intellect Books, oxford, England. 1993

[5] Johann Gamper, Judith Knapp. A Review of Intelligent CALL Systems. Computer Assisted Language Learning (CALL): An International Journal, 15(4), 329-342, Belgium: SWETS & ZEITLINGER publisher, 2002.

[6] Merryanna L Swartz, Masoud Yazdani. Intelligent tutoring systems for foreign language learning, chapter Introduction. Springer-Verlag, 1992

[7] Mourad Mars, Georges Antoniadis, Mounir Zrigui. Which Algorithm and Approach for Arabic Part Of Speech Tagging. CITII 2010. 27-29 octobre 2010, (Journal Research in Computer Science), Apizaco, Mexique. 2010

[8] Mourad Mars, Georges Antoniadis, Mounir Zrigui. Statistical Part of Speech Tagger for Arabic Language. ICAI'10 - The 2010 International Conference on Artificial Intelligence. 12-15 juillet 2010, Las Vegas, Nevada, USA. 894-899. 2010

[9] Mourad Mars, Georges Antoniadis, Mounir Zrigui, "Nouvelles ressources et nouvelles pratiques pédagogiques avec les outils TAL", TICEMED 08, Journal Information Sciences for Decision Making (Journal ISDM), ISDM32, N°571, Avril 2008.

[10] Mohsen Maraoui, Georges Antoniadis, Mounir Zrigui: CALL System for Arabic Based on Natural Language Processing Tools. IICAI 2009: 2249-2258.

[11] Nielsen H. and Carlsen M. Interactive Arabic Grammar on the Internet: Problems and Solutions, Computer Assisted Language Learning (CALL): An International Journal, Vol. 16(1):95-112, SWETS & ZEITLINER publisher. 2003

[12] Olivier Kraif , Georges Antoniadis, Sandra Echinard, Mathieu Loiseau, Thomas Lebarbé, Claude Ponton . NLP tools for CALL: the simpler, the better. In Proceedings of InSTIL/ICALL2004 Symposium, NLP and Speech Technologies in Advanced Language Learning Systems. 2004

[13] Shaalan F. Kaled.: An Intelligent Computer Assisted Language Learning System for Arabic Learners, Computer Assisted Language Learning: An International Journal, 18(1 & 2), PP. 81-108, Taylor & Francis Group Ltd., February 2005.

[14] Shaalan Khaled F., Habib E. Talhami. Arabic Error Feedback in an Online Arabic Learning System1. Advances in Natural Language Proccesing, Research in Computing Science 18, pp. 203-212. 2006

[15] Sherry, L. Issues in distance learning. International journal of Educational telecommunications, Vol. 1, No. 4, pp. 337-365. 1996

[16] Stenerson, j. System Analysis and Design for a Successful Distance Education Program Implementation. Journal of distance learning administration, Vol. 1, No. 2.1998

[17] Taylor, J. Fifth Generation Distance Education, In the Proceedings of the 20th world Conference on Open learning and Distance Education, International Council for Open and Distance Learning, ICDE, Usseldorf, Germany. 2001

[18] Thorandike, R., Cunningham, G., and Hegen, E. Measurement and Evaluation in Psychology and Education. New York: Macmillan Publishing Company. 1991

[19] Wiersma, W., and Jurs, S. Educational Measurement & testing. Boston: Allyn and Bacon. 1991

The analysis of print media discourse in the election context

Daniela Gîfu
Faculty of Computer Science, „Alexandru Ioan Cuza“ University, Iaşi, Romania

Abstract - *This paper presents a computational method, AnaDiP-2011, based on natural language processing (NLP) techniques for the interpretation of the political discourse in the print media. The application considers the 2009 presidential campaign in Romania. The concept behind this method is that the manner in which individuals speak and write, with the aim to deliver a certain image to the public, is an opened window towards their emotional and cognitive worlds. By emphasizing the emotional component at the level of discourse, voters identify with the speaker, who becomes the personification of their common ideals. Our investigation is intended to give support to researchers, specialists in political sciences, political analysts and election's staff, being helpful mainly in their social exploration of the electoral campaigns in their intend to measure reactions with respect to the developments in the political scene*.

Keywords: political discourse, natural language processing, newspaper, semantic analysis, elections

1 Introduction

It is known that the text of the press may be of inconvenience or can be useful to those that become the subject of the press. Most of the time the subjects are the ones in political power of that time. "In contemporary postindustrial society, the conflict stake is no longer the social use of technology, but mass production and dissemination of representations, information and languages" (Touraine, 1992: 184)

The motivation for our study relies on the need for objectivity in the interpretation of the political discourse, situated at the intersection of three important symbolic spaces: the political space, the public space and the communicational space (Wolton, 1998), as well as on the need to measure to what extend a discourse can influence its direct receptor, the electorate, and in what ways. The political language, essential in building a discourse (Perlmutter, 1999), for print media discourse in the election context also, requires an interdisciplinary approach in which linguistics should co-ordinate with journalism, the rhetorical science, the communication sciences, the political sciences and the sociology.

Among many attributes the political discourse has, we were interested in the lexicon and its interpretation in a range of semantic coordinates. The final objective of our research is to develop a computational framework able to offer to the researchers in to mass-media, political sciences, to political analysts, to the public at large (interested to consolidate their options before elections), and, why not, even to politicians themselves, the possibility to measure different parameters of a written political discourse. Based on these parameters they should be able to appreciate certain aspects characterising the author of the discourse, as shading lights on his/her personality, or on the way he/she perceives the society or only some levels of it, as well as on his/her persuasive arsenal, etc. This can be done provided we will be able to link the statistical values outputted by the computational tool onto facts about the author of the message or the reality he/she is depicting. We were aware that the interpretation of numerical findings the program outputs should be validated by human experts in order to become facts. Part of our research, as reported in (Gîfu, 2010), was concentrated on this type of human validation.

The software we developed, AnaDiP (Political Discourse Analysis) offers the possibility to analyze efficiently large bodies of text and to characterize them quantitatively and qualitatively, the results having to be as close as possible to the analysis made by a human expert. The system offers a global perspective over the political discourse, as well as a punctual one.

The paper is structured as follows. Section 2 shortly describes the functionality of the software and the associated resources for the Romanian language and some suggestions about the words disambiguation. Then, section 3 discusses two example picked up from the print media (the editorials, the political discourses) during the 2009 Romanian presidential elections, section 4 highlights several challenges and sections 5 presents the conclusions.

2 The software and lexical resources

Although, as mentioned, the functionality of AnaDiP is inspired by LIWC (Pennebaker, James W., Francis, Martha E., Booth, Roger J., 2001), there are important differences between the two platforms. LIWC-2007[1] is basically counting words and incrementing counters of all their declared semantic classes, when they are discovered in the input text. In the lexicon, words can be given by their long form, as a complete string of characters, or abbreviated, in which case

[1] www.liwc.net

the sign '*' plays the role of the universal jolly-joker, replacing any character. For each text in the input, LIWC produces a set of tables, each displaying the occurrences of the word-like instances of the semantic classes defined in the lexicon, as sub-unitary values. For one semantic class, such a value is computed as the number of occurrences of the words corresponding to that class divided by the total number of words in the text. It remains in the hands of the user to interpret these figures. And there is no support for considering lexical expressions.

We will refer now to the way in which AnaDiP organizes the lexicon and how it counts words. The software performs part-of-speech (POS) tagging and lemmatization of words. This is why the lexicon can now be declared as a collection of lemmas having the POS categories: verb, noun, adjective and adverb. As seen, we leave out the pronouns, numerals, prepositions and conjunctions, considered to be semantically empty. An entry of the lexicon has the form: <lemma> <POS> <sem-list>, where <sem-list> is a list of semantic classes. This means that the same lemma can appear with more than one POS and, if needed, with different semantic interpretations.

Although the introduction of lemmatization and POS tagging makes useless the 'any' operator, we have kept it in the definition of the lexicon entries. The user has the possibility to either define an entry as a <lemma><POS><sem-list> triple, as explained above, or as <word-root>(*) - <sem-list>, with the significance that the root can be ended with a '*' sign, no POS is defined and any word matching the root during analysis will increase the counters of all semantic classes belonging to the <sem-list>.

The second range of differences between the two platforms stays in the user interface. In AnaDiP, the user has an easy to interact interface, offering a lot of services: opening of one or more files, displaying the file/s, modifying/editing and saving the text, functions of undo/redo, functions to edit the lexicon, visualizing the mentioning of instances of certain semantic classes in the text, etc. Then, the menus offer a whole range of output visualization functions, from tabular form to graphical representations and to printing services. Figure 1 shows a snapshot of the interface during a working session.

Finally, another important development was the inclusion of a collection of formulas which can be used to make comparative studies between different subjects. In section 3 we will present an example.

The Romanian lexicon of contains now approximately 6000 words and roots and 28 semantic classes[2]. We plan to populate our lexicon further by importing from DEX-online[3], the greatest public online dictionary for Romanian. The semantic classes in AnaDiP are partially placed in a hierarchy. In the future we plan to align this hierarchy with WordNet (Fellbaum, 2001), for languages which support this type of linguistic resource.

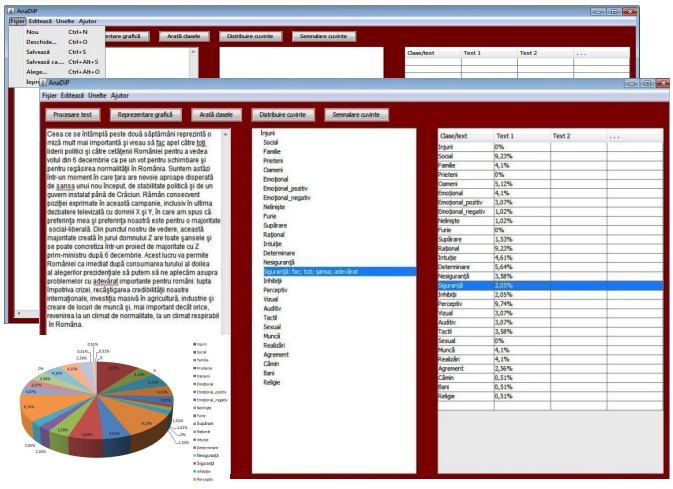


Fig. 1. The AnaDiP interface: in the left window appears the selected text, on the middle window – the classes, and in the right window – the words and their respective counted percentages and the chart, chosen by the user (here, type pie). By selecting some word on the right or a class in the middle window, the corresponding occurrences in the text are highlighted in the left window.

A special section of the lexicon includes expressions. An expression is defined as a sequence: <root-list> => <sem-list>, in which <root-list> is a list of roots of words, therefore each optionally followed by the '*' sign. Because, in principle, a root can also be a numerical value and the semantic classes in <sem-list> are indicated by numbers, to separate the roots section from the semantic categories section we had to place a special sign (=>). Each time a sequence of words matching the <root-list> is recognized in the text, the counters associated with the semantic classes in the <sem-list> are increased.

As an example, for the root plictico*[4], the following classes are assigned: 6 = Affect, 8 = Negative_emotion and 11 = Sadness. Whenever the word or a variant of the word is detected in the input file, all three counters, corresponding to the classes mentioned, are incremented.

The 28 semantic classes included now in AnaDiP (Table 1) have been selected to fit optimally with the necessities to interpret the political discourse of the presidential campaign in Romania, in 2009. We were mainly interested to determine those political attitudes which were able to influence the

[2] These classes were selected from 64 classes (syntactic and semantic) that are included in the software LIWC-2007, original version. Taking into account the criterion of persuasion, which each stage of political discourse builds, we considered the 28 semantic classes necessary and sufficient for our study. Thus, the expected purpose will show the connection between the predominant proportion of classes and voting options. Note that the codes were changed as the American package software and class names are translated into Romanian.

[3] www.dexonline.ro

[4] from *plicticos*, *plicticoasă*, *plicticoşi*, *plicticoase* (*boring*)

voting decision of the electorate. However, the user can define at his/her will these classes and the associated lexicon.

code	Class in Romanian (English)	Description and examples in RO (EN)
1	Injurii (Swear)	Words that harm the reputation of someone: afurisit, cretin, jigodie (damned, idiot, cur)
2	Social (Social)	Words about social appurtenance: preşedinţie, eveniment, celebrare (presidency, event, celebration)
3	Familie (Familiy)	clan, părinte, fiu (clan, parent, son)
4	Prieteni (Friends)	camarad, coleg, companion (comrade, colleague, companion)
5	Oameni (Humans)	adult, personalitate, ţăran (adult, personality, peasant)
6	Emotional (Affect)	Words with emotional effect: abătut, curajos, impasibil (depressed, courageous, impassive)
7	Emotional_pozitiv (Positive_emotion)	Words with positive effect: ademenitor, plăcut, popular (alluring, nice, popular)
8	Emotional_negativ (Negative_emotion)	Words with negative effect: agresiv, cinic, trândav (aggressive, cynic, slothful)
9	Anxietate (Anxiety)	alarmant, tensionat, bănuitor (alarmed, jumpy, suspicious)
10	Furie (Anger)	brutal, ofensator, irascibil (brutal, offensive, irascible)
11	Supărare (Sadness)	îngrijorător, revoltător, trist (alarming, revolting, sad)
12	Raţional (Rational)	inventiv, discernământ, precaut (inventive, discernment, precautious)
13	Intuiţie (Intuition)	ager, explicativ, realist (astute, explicative, realist)
14	Determinare (Cause)	deductiv, insinuant, manipulator (deductive, insinuate, manipulator)
15	Nesiguranţă (Uncertainty)	vag, confuz, ambiguu (vague, confused, ambiguous)
16	Siguranţă (Certainty)	exact, hotărât, categoric (precise, resolute, absolutely)
17	Inhibiţie (Inhibition)	absent, descurajat, şovăitor (absent, discouraged, giddy)
18	Perceptiv (Percept)	Words related to perceptions: auzi, observa, palpabil (hear, observes, tangible)
19	Vizual (See)	holba, lumină, orb (stare, light, blind)
20	Auditiv (Hear)	acustic, tăcut, surd (acoustic, silent, deaf)
21	Tactil (Feel)	ascuţit, fragil, estompat (sharp, fragile, blurry)
22	Sexual (Sexual)	Words with a sexual tonality: pipăi, dezmierdare, încins (caress, endearment, hot)
23	Muncă (Work)	Work specific words: calificare, department, organizaţii (qualification, department, organization)
24	Realizări (Achievements)	Words that reveal human achievements: facultate, succes, competenţă (college, success, skills)
25	Agrement (Leisure)	Words from the recreational domain: dans, cazinou, camping (dance, casino, camping)
26	Cămin (Home)	Words related to house: cameră, decor, mansardă (room, decoration, attic)
27	Bani (Money)	Words related to economy and finance: avere, cont, licitaţie (wealth, account, auction)
28	Religie (Religion)	Words with a religious tonality: Biblie, divinitate, creştin (Bible, divine, Christian)

Table 1. The 28 semantic classes currently included in the AnaDiP package

Keeping in mind the remarkably sophisticated and time consuming process in which the Romanian version of the dictionary LIWC-2007 was acquired, we knew that some decisions have to be taken in order to optimize its content while also diminishing the influence of English, evident if a simple EN-onto-RO translation process would have been applied. The development of the lexicon was done in several phases:

- first the LIWC-2007 English terms belonging to the 28 classes previously mentioned, considered meaningful for this type of analysis, were translated, retaining only the Romanian words which had senses in accordance with the corresponding classes;

- then, words / roots in each class were sorted alphabetically;

- then we have reconsidered each class in part, eliminating words that could introduce ambiguities and including synonyms. We have done this activity with a class of master students in Computational Linguistics[5]. Then, the work done by students has been once again validated by both authors.

[5] In 2010, in their 1st year at the Faculty of Computer Science, the "Alexandru Ioan Cuza" University of Iaşi.

By working within one semantic class at a time, we were able to easily recognize classification mistakes and correct them. Also, the alphabetical ordering offers the possibility to operate certain roots optimizations, by exploiting the use of the jolly-joker *;

- then a close look in a part of the documents. The collection was done by one of the authors. When a competent reader does this, normally do pop-up important words. They were checked against the DEX-online lexicon by using '*' to include more possibilities and the resulting matches were included in the AnaDiP-2011 lexicon.

- then, the dictionaries assigned to all classes were merged, and the obtained list was sorted again alphabetically. Multiple occurrences were removed (by clashing together the lists of classes, and leaving only one instance of a class in the list corresponding to one root/word);

- then, when the root/word notation (including or not the use of '*') was seen to give rise to unwanted ambiguities, the <lemma><POS> notation was used instead;

- finally, starting from words, expressions were introduced;

- as mentioned already, in a future stage, we will use the words we have now in the lexicon as seeds in a trial to enrich it automatically, by making use of DEX-online. We study now strategies to exploit the synonymy relation and the definitions.

2.1 Word connotations disambiguation

One of the problems that face us right now is: *How do we establish which of the senses of an ambiguous word is invoked in a particular use of the word?* Because, each part of press discourse, especially political discourse, is constructed using a lot of connotations for the same word. In other words, how we resolve the task of disambiguation to all words from the lexicon. This is done by looking at the context of the word's use. Here, electoral context. A word is assumed to have a finite number of connotations, often given by a dictionary or other reference source. The task of the program is to make a forced choice between these connotations for the choosing the right sense of word, based on the context of use. However, it is hard task to realized. A few examples, like the word *bank* and much more we find in literature (Manning, Christopher D., Schütze, Hinrich, 2003).

One approach is simply to define the senses of a word as the meanings given to it in a particular dictionary. However, this is unsatisfactory from a scientific viewpoint because all reference sources often differ greatly in the number and kind of connotations they list. A pattern of sense extension between a concept and something that shows the concept is pervasive and could have been, but was not, distinguished for other uses. For example the same ambiguity exists when talking about public politics. For instance, one might remark in a politic debate: *These public politics don't have a title*. *With what their will improve people's lives*? These difficulties suggest that, for most words (e.g. public politics), the usages and hence the sense definitions are not to be thought of like "n" kinds of pieces to something, among which one must choose, but more like a unit which has some pieces of clearly distinct indentifiable content, but a lot of stuff of uncertain and mixed origin in between.

Notwithstanding these philosophical objections, the problem of disambiguation is of clear importance in many applications of natural language processing. A system for automatic translation from English to Romanian needs to translate closely the word sense, but it is impossible for the moment (e.g. two examples of section 2.2.). It depends on the context, linguistics, cultural values etc. Whenever a system's actions depend on the meaning of being processed, disambiguation is beneficial or even necessary.

2.2 Lexical-based disambiguation

We started from Lesk's experiences (Lesk, 1986: 24-26) and from suggestion by Pook and Catlett, also (Pook and Catlett, 1988: 148-157). He exploits the sense definitions in the dictionary directly. His idea is that a word's dictionary definitions are likely to be good indicators for the senses they define[6]. We describe a disambiguation method that rely on the definition of senses in two dictionary. We considered two definitions for institution from both dictionaries, as follows:

a) DEX-online, a very popular Romanian dictionary.

1. an unity with political, economic, administrative... profile which perform a specific activity.
2. organization which perform an international interest of activity.

or

b) WORDNET[7], a lexical database for the English language.

3. an organization founded and united for a specific purpose.
4. a custom that for a long time has been an important feature of some group or society.

If either *unity* or *organization* (first example) and *organization* or *custom* (second example) occur in the same context as *institution*, then chances are that the occurrence belongs to the sense whose definition contains that word:

- sense 1 for *unity*, sense 2 for *organization*.
- sense 1 for *organization*, sense 2 for *custom*.

Let $D_1,...,D_k$ be the dictionary definitions of the connotations $s_1,...,s_k$ of the ambiguous word w, represented as the bag of words occurring in the definition and E_{v_j} dictionary definition of a word v_j occurring in the

[6] Lesk credits Margaret Millar and Laurence Urdang with the original proposal of the algorithm.

[7] wordnet.princeton.edu

context of use c of w, represented as the bag of words occurring in the definition of v_j. (If $s_{j_1},...,s_{j_i}$ are the senses of v_j, then $E_{v_j} = U_{j_i} D_{j_i}$. We simply ignore sense distinctions for the words v_j that occur in the context of w.) So, the Lesk's algorithm can be described as shown in next algorithm:

1. **comment**: Given context c
2. **for** all senses s_k
3. $score(s_k) = overlap(D_k, U_{v_j\ inc} E_{v_j})$
4. **end**
5. **choose** s' s.t. $s' = \arg\max_{s_k} score(s_k)$

where: D_k is the set of words occurring in the dictionary definition of sense s_k; E_{v_j} is the set of words occurring in the dictionary definition of word v_j (that is, the union of all the sense definitions of v_j).

Anyway, information of this sort derived from a dictionary is insufficient for high quality word sense disambiguation. However, Lesk reports accuracies between 50% and 70% when the algorithm is applied to a sample of ambiguous words.

We work now to the different optimizations that could improve performance. First, we have in attention 2 or 3 textual corpuses on the various themes (political, literature, social). Then, we could run several iterations of the algorithm on each text. Instead of using the union of all words E_{v_j} occurring in definition of v_j, we could only use the words in the definitions of the contextually appropriate senses as determined in the previous iteration of the algorithm. We should hope that the iterated algorithm eventually settles on the correct sense of each word in the text. Of course, we have in attention the context of discourse and all synonyms for each word which the software counts. We have for the moment a list with these synonyms.

3 The analysis of print media discourse during the 2009 Romanian presidential elections

For the elaboration of preliminary conclusions over the presidential elections process, conducted in the period 23 October – 4 December 2009 in Romania, we collected, stored and processed electronically, in three different stages (one month before, the 1st tour and, finally, the 2nd tour of the election's campaign), political texts, i.e. editorials published by three national publications having similar[8] profiles, as well as political speeches (in both oral and written form), all belonging to four candidates for the function of President.

The monitored written media corpus was processed directly with AnaDiP. The speech records were previously manually transposed onto text and then they followed the same processing as the written texts. In essence, the program receives the input from one or more text files, and counts occurrences of words belonging to its defined classes. The user can notice directly the mentioned semantic classes (and the corresponding frequencies), as the words belonging to a selected class appear underlined in the left screen. The user can choose a type of graphical representation ("function", "pie" or "columns"), which give intuitive visual perceptions on which the interpretation of discourse data can be performed more conveniently.

Apart from simply computing frequencies, the system can also perform comparative studies. The assessments made are comprehensive over the selected classes because they represent averages on collections of texts, not just a single text.

AnaDiP provides a library of comparative functions, with 2 to 4 different input streams of data. One stream can be either a newspaper, or only one discursive approach on a certain topic delivered at a certain moment in time by the traced author.

3.1 The analysis of editorials

To exemplify, one type of graphics considered for the interpretation was the one-to-two difference, as given by Formula (1), included in the AnaDiP Mathematical Functions Library:

$$Diff^{1-1}_{x,y} = average(x) - \frac{average(y) + average(z)}{2} \quad (1)$$

where x and y and z are three streams; average(x), average(y) and average(z) are the average frequencies of x, y and z over the whole stream, and the difference is computed for each selected class. Since a difference can lead to both positive and negative values, these particular graphs should read as follows: values above the horizontal axis are those prevailing at the daily x versus the daily y and z, and those below the horizontal axis show the reverse prominence. A zero value indicates equality.

To exemplify, we present below a chart with two streams of data, representing the editorials of the second tour of voting.

[8] The three newspapers have been *Evenimentul Zilei*, *Gandul* and *Ziua* (www.mediapres.ro), which are known to have a common profile: national dailies of general information, tabloids with a circulation of tens of thousands of copies per edition, each. The newspapers were monitored on their websites: www.evz.ro, www.gandul.info, www.ziua.ro.

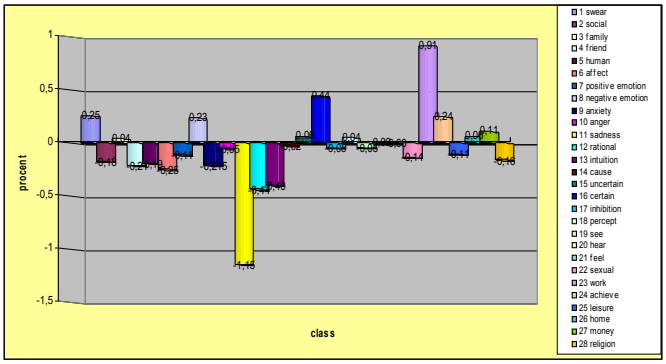


Fig. 2. The average differences in the frequencies for each class after processing the editorials (the second tour), between the newspapers: *Evenimentul Zilei* versus *Gândul* & Ziua.

Our experience shows that an absolute difference value below the threshold of 0.5% should be considered as irrelevant and, therefore, ignored in the interpretation. So, the graphical representation in Fig. 2, in which *Evenimentul zilei* is compared with *Gândul* & *Ziua*, should be interpreted as follows: *Evenimentul zilei*'s discursive intervention has an interest more towards the working aspects (Work) than *Gândul* and *Ziua* together, which had a negative emotional (Sadness) attitude.

3.2 The analysis of political discourses

To exemplify, another type of graphics considered for the interpretation was the one-to-one difference, as given by Formula (2), included in the AnaDiP-2011 Mathematical Functions Library:

$$Diff_{x,y}^{1\text{-}1} = average(x) - average(y) \quad (2)$$

where *x* and *y* are two streams; *average(x)* and *average(y)* are the average frequencies of *x* and *y* over the whole stream, and the difference is computed for each selected class. Since a difference can lead to both positive and negative values, these particular graphs should read as follows: values above the horizontal axis are those prevailing at the candidate *x* versus the candidate *y*, and those below the horizontal axis show the reverse prominence. A zero value indicates equality.

To exemplify, we present below a chart with two streams of data, representing the texts of the final TV debate between the two remaining presidential candidates in the election race, the second tour of voting.

In the graphical representation of Fig. 3, we compare Traian Băsescu (TB – a Democrat Liberal Party candidate) against Mircea Geoană (MG – the Social Democrat Party), by applying Formula (1): as such, if a value of a class is positive (above the horizontal axis) it shows the prevalence of TB over MG in that class, and if it is negative (bellow the horizontal axis), it shows the prevalence of MG over TB in that class. This should be interpreted as follows: TB's discursive intervention are much more `perceptual` than MG's, therefore touching directly onto human common sense, being also rather careful (`Inhibition`), while his opponent, MG, has touched more onto social issues (`Social`), quite normal, seen his affiliation with Social Democrats, had an emotional attitude (`Affect`), and with an obvious interest towards the working aspects (`Work`).

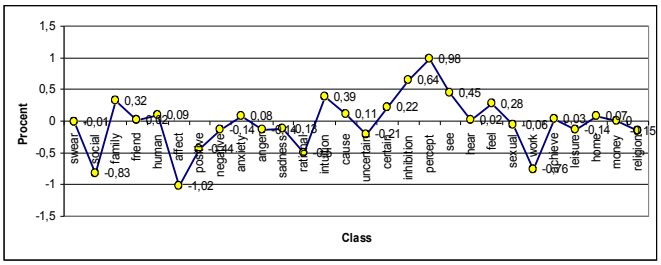


Fig. 3. The average differences in the frequencies for each class after processing the final TV debate, December 4, 2009, between the candidates: Traian Băsescu versus Mircea Geoană.

Many of the conclusions found by the program have been confirmed by political commenters. Moreover, the program helped also to outline distinctive features which brought a new, and sometimes even unexpected, vision upon the discursive characteristics of the journalists, of the presidential candidates and, last but not least, of the Romanian voters, at the end of 2009.

4 Discussion: Challenges

The first challenge arises from interpretation of what constitutes persuasion. Half-justifications and half-emotions inevitably interfere with binary classification by journalists / political candidates. In general, journalists / political candidates performed better when their themes (e.g., the 2009 Romanian presidential elections) were treated with excessive emotional tonalities. In this case, the lectors can be manipulated easily (e.g., Sadness class). It would be beneficial if humans could defer to a machine learning prediction in cases where they lack confidence.

Second, our data elicitation task was opened, allowing all analysts, journalists, communication specialists to make choices and us to present their discourse preferences in function about their interests. The heterogeneous data, however, lessens linguistic predictive power. Previous computational attempts succeeded with well-defined tasks for discovering working aspects (e.g., Work class) which becomes the most significant deception in human's life (Hanckock et al., 2008: 1-23) or an opinion entirely contrary to one's belief (Mihalcea and Strapparava, 2009: 309-312). Also, following the qualitative interpretation of different types of emotion levels, we proposed to analyze the election

context in function to the few categories: public agenda, candidates or a focal point of the political scenarios.

Finally, in our study objectives we may be interested how much the journalists / political candidates have prompted them to apply extra effort in "fooling" receptors with yet unbelievable editorials / political discourses. In other words, how much rational and how much emotional.

5 Conclusions

Currently, the NLP domain has reached a scientific and technological maturity that makes it useful in the activities carried out by researchers in socio-humanity fields, and initiatives like CLARIN[9] show that a consistent bridge between these two communities can be built.

We believe that AnaDiP has a range of features that make it attractive as a tool to assist political campaigns. It can also be rapidly adapted to new domains and to new languages, while its interface is user-friendly and offers a good range of useful functionalities.

In the future we intend to include a word sense disambiguation module in order to determine the correct senses, in context, of those words which are ambiguous between different semantic classes belonging to the lexicon, or between classes in the lexicon and outside the lexicon (in which case they would not have to be counted).

Acknowledgments: We thank professor Dan Cristea, from the Faculty of Computer Science of the "Alexandru Ioan Cuza" of Iaşi, for initiating this research and for supervising the author, as a post-doc researcher. The AnaDiP software has been developed by Mădălina Spătaru from the Faculty of Computer Science "Alexandru Ioan Cuza" of Iaşi. The master students in the 1st year of Computational Linguistics at the Faculty of Computer Science of the "Alexandru Ioan Cuza" of Iaşi have participated in the initial phases of the acquisition of the lexicon. In order to perform this research the author received financial support from the POSDRU/89/1.5/S/63663 grant.

6 References

[1] Christiane Fellbaum (ed.): "WordNet, An Electronic Lexical Database", The MIT Press, 2001.

[2] Daniela Gîfu: "The Discourse of the Written Press and the Violence of Symbols" (in Romanian), PhD thesis, Faculty of Philosophy and Political Studies, "Alexandru Ioan Cuza" University of Iaşi, 2010.

[3] J. T. Hancock, L. E. Curry, S. Goorha and M. Woodworth: "On lying and being lied to: A linguistic analysis of deception in computer-mediated communication." Discourse Process 45 (1).

[4] Michael Lesk: "Automatic sense disambiguation> How to tell a pine cone from an ice cream cone. In *Proceedings of the 1986 SIGDOC Conference*, New York, Association for Computing Machinery.

[5] Christopher D. Manning, Hinrich Schütze: "Foundations of Statistical Natural Language Processing", London, England, The MIT Press Cambridge, 2003.

[6] R. Mihalcea and C. Strapparava: "The Lie Detector: Exploration in the Automatic Recognition of Deceptive Language". Proceedings of the Association for Computational Linguistics (ACL 2009).

[7] James W. Pennebaker, Martha E. Francis, Roger J. Booth: "Linquistic Inquiry and Word Count – LIWC2001", Mahwah, NJ, Erlbaum Publishers, 2001.

[8] David D. Perlmutter: "The Manship School guide to political communication", Baton Rouge, Louisiana State University Press, 1999.

[9] Stuart L. Pook, Jason Catlett: "Making sense out of searching". In *Information Online 88*, Sydney. The Information Science Section of the Libraru Association of Australia, 1988.

[10] Alain Touraine, "Critique de la modernité", Fayard, Paris, 1992.

[11] Dominique Wolton: "Les contradictions de la communication politique", CNRS, Paris, 1995.

[9] www.clarin.eu

Interaction Matrix Model for Language Production

Steven Gibson
Department of Communications Studies, California State University, Northridge, USA

Abstract— *The Congruent Interaction Matrix (CIM) model is being formulated to represent knowledge updating and language production. The line of research taken in this paper is limited to a set of questions restricted to concepts and actions that can be used in modeling human language behavior. The Congruent Interaction Matrix model introduced here proposes virtual structures represented as matrices. The theoretical and practical value of developing this framework and set of algorithms is discussed, in order to create tools useful for modeling human communication interactions. Possible future research studies and applications are suggested. The development of these tools could have future implications for human and machine communication analysis and production.*

Keywords: knowledge representation, model, framework, communication, automated reasoning.

1. Introduction

Computer generation of human language is now in common use. Improvements in computer language production may produce more diverse output through the application of new concepts and algorithms. The construction of a Congruent Interaction Matrix (CIM) virtual model could be useful in representing communication behaviors. The CIM Model is composed of a data structure, information and a set of rules to mediate interactions. The structure is implemented as matrices which are composed of cells that can include information enmeshing individual interaction details. These components, in the form of cells, also represent ways to recall stored meaning. Matrix components can link to each other based on content and meaning; each CIM component may contain a rule which may actuate and then result in that component becoming part of the one active matrix. During communication interactions, the active matrix results from the rules being actuated. In this way, a model can be built in which individual rules interact during simulated or real interactions. The matrix experiences interaction results when the matrix is reorganized by the actuation and matching of components.

This paper explains the design and implementation of software that can model an aspect of language use. The software postulates that human linguistic activities can be treated as matrices that can be matched, recalled and replaced. The model represents activity of matrices in the production of language behaviors. During a simulated run, the model represents the interpretation of sensory data and the responding language behavior. This model is proposed to offer a realistic approach to language production while the software demonstrates matrix interactions used in knowledge and language production.

2. Research Question

There is value in building a mathematical or algorithmic model for human speech production. This model would need to represent how individuals choose words and sentences in response to input. Aspects of the model then need to demonstrate how to generate new language or new knowledge. This system for modeling language production should be testable with simulations or mathematical modeling and it seems likely that humans experience knowledge updating in a way that can be modeled by computers [1]. The expected outcome is the construction of a valuable research tool that will serve in studies of artificial and natural language production. Inspiration for this approach to language comes from data-driven approaches to language generation like VINC [2].

3. Interaction Matrix

The CIM model can be described in different ways. This section provides descriptions of the mathematical and algorithmic model in a Congruent Interaction Matrix. Two key aspects of the model are the data structure approach to interaction components and the process of interactions actuating new components. This model assumes that an individual communication interaction uses stored data and can undergo knowledge updating and change during its lifetime. A list of aspects of the Interaction Matrix model includes a collection of data points, response cues and programmed components. Statistical rules as well as single cases should be equally valid with this model. Individual cases are first examined, although the model may also be equally valid for groups.

Because every individual is unique, variation is normal; interaction matrix properties shift because individuals have built in knowledge and respond to new input through language. There is a natural variation in the internal knowledge of individuals and their responses. In individual cases, under examination, the internal state contains the components and any other knowledge relevant for engaging in verbal behavior. A complete Interaction Matrix process is intended to model language behavior in the world and will involve

perception, conceptualization, component lookup, parsing and speech production.

The process for a language actor consists of a two-way association between input components (individual words and other components) and internal components (words, meanings, and connections). Each matched association will generate a score. When an actor needs to process components, she looks up all possible component matches, orders them and picks the one with the best score for generation of a new matrix. Thus the resulting matrix uses the components with the highest score as elements in the active matrix. New meaning is defined in the process through the choice of components.

3.1 Spreadsheet Representation

One metaphor to use in visualizing this model is that of a spreadsheet [3]. The active matrix can be equated to a column in a spreadsheet with future potential matrices viewed as additional columns. Each component that is part of the communication is represented as a cell in the column. The active matrix is updated in response to a match between input with potential matrices.

Table 1: Sample CIM spreadsheet format.

	Active	To Merge	1st Potential
Match Words	hello	hello	goodbye
Output Words	hello friend	hello	goodbye friend
Emotion	happy	joy	fear
Relationship	friend	enemy	stranger
Frame	home	business	game
Countdown Timer	2	3	2
Congruence Value	na	2	na

The input column is not shown in table one, but the input models what the individual perceives and constructs from sensory input. The components in the input are matched to other columns to result in a new active column. Additional columns then will be added to the right of the 1st column, with subsequent columns following in the same way (i.e., 2nd, 3rd, etc.)

3.2 Pseudocode Representation

Here is a high-level description of one iteration of a simulated run of the computer version of the model:

```
process input from user interface (1)
begin timer stopwatch based on active
 column value (2)
perform search on 1st column in queue (3)
  until at least one element matches
  continue search for additional matches
  return success if at least one match
  occurred
repeat search on succeeding columns
  until out (4)
  of columns or timer runs down
When timer runs out (5)
   if at least one successful column
   was returned, make it active
   if no successful column returned
   do not change active
if active was changed return sentence (6)
   prepare to process new input (1)
```

Note: This pseudocode version does not retain execution order or data structure.

3.3 Text description

The software system is broken down into modules to handle aspects of the task and serve to process the submitted cues and send out requests for matches. Each possible match is returned by a module and kept in a buffer until the stop clock tics down. Better matches continue to be sent to the buffer until the timer runs down. Once that occurs, a kill message is sent to all remaining modules to stop the searching. Each possible match that reaches the buffer is compared by quality with the previous match of this series; the better match is retained while the other is discarded. After the timer runs out, the successful match is moved to the top of the queue for subsequent searches.

4. Applications

This model has the potential to serve in A.I. applications to aid in language research and as a means to examine knowledge production [4]. The model can aid in exploring how to track effective decision selection. Repeated modeling of sample language production outputs may help in understanding human language use. Furthermore, theories and practices can be compared and contrasted through the use and modification of an adequate communication model. Through these processes, steps may be taken to arrive at new research programs for linguistics which may aid in explaining the characteristics and interactions between actors, ultimately serving to model and produce possible concepts of language evolution.

5. Computer Simulation

Part of the development of this model includes a software component to aid in feasibility studies of language production. Requirements include concurrent searches for matches, input of data and result output. Erlang, a general-purpose concurrent functional programming language with strict evaluation, single assignment and dynamic typing, was the software system chosen to construct the model [5]. This system takes input which consists of words along with emotional, relational and social cues. The software developed (ERDlanguage), operates through a web interface to produce sentences in response to input. The software

consists of a multi-module server backend that interacts with a web browser-based interface for human input and computer output.

5.1 Software

The software will run anywhere Erlang can run, including Linux, Windows and Mac computer desktops and laptops. A web browser serves as the front end interface for input. The server backend can run on another machine on the network or accessible over the internet. The software package consists of Erlang source code files, multi-language interface text, and a data file that holds possible input and responses.

To download the software from this project visit:
http://sourceforge.net/projects/erdialog/files/erdlanguage/
An online version is located at: http://stevengibson.org:8101

5.2 Screen

A screenshot is included here of the browser-based interface for the software.

Fig. 1: Sample data screen.

6. Discussion

In this paper, a new model (CIM) has been advanced which is constructed with a mathematical and algorithmic structure. After initially providing a summary and boundary framework for the model, likely test beds were described where the model could be implemented and utilized and a working implementation is produced as a complete software product. Some possible benefits for developing this model are listed. Additional aspects of a demonstration have been offered through the application of an example case. It is possible that this model could be applied to human language production and A.I. applications.

The Congruent Interaction Matrix model could be tested in a heuristic manner while providing theoretically meaningful test results. Future CIM models could be produced in evolutionary ways similar to how linguistics describes language change. One possible line of research could use this system to develop an explanatory framework for modeling language production, and in this way capture a richer range of human communication interaction. Part of this model assumes that the study of human communications interaction requires building models based on actors whose internal states reflect a more diverse complexity of interaction components which can be modeled. Experiments need to include techniques and technologies that do language simulations and experiments to help develop corrections to the CIM model. The plan is not to implement CIM in one frozen state but in versions and releases that adapt continuously to the environment of human or artificial actors.

Possible significant contributions from this work are likely in the field of communications studies where queries can be carried out in modeling human verbal dialog. Natural language production by computer has involved development of rules and data sets that produce meaningful output. And while software development has achieved success, more efforts are in order. The rule and data set approach produced inconsistent results and does not provide any metalinguistic judgments of either pragmatic or contextual variation in language production [2]. This present software offers a first step in supporting language production.

One approach to the process of constructing natural language-like outputs from human-like linguistic inputs is shown here. A limitation of the approach used here is the use of simple one word inputs. While the method added the idea of including emotional, relational and framing cues, the input did not include valid sentences. Future steps should involve the input and processing of valid human originated sentences to generate the language outputs.

References

[1] S. Gibson, “Process algebra modeling of human communication," pp. 685–688, In H. R. Arabnia and D. de la Fuente and J. A. Olivas, editors, CSREA Press, July 2009.

[2] M. Levison and G. Lessard, “A system for natural language sentence generation," *Computers and the Humanities*, vol. 26, no. 1, pp. 43–58, 1992.

[3] S. Powell, K. Baker, and B. Lawson, “A critical review of the literature on spreadsheet errors," *Decision Support Systems*, vol. 46, pp. 128–138, Dec. 2008.

[4] S. O. Hansson. “A survey of non-prioritized belief revision," *Erkenntnis (1975-)*, vol. 50, no. 2/3, pp. 413–427, 1999.

[5] J. Armstrong. “Erlang,” *Commun. ACM*, vol. 53, pp. 68–75, September 2010.

Research on Rule-based Chinese Syntactic Parsing Postprocess Using Verb Subcategorization

Jinyong Wang[1], **Xiwu Han**[2]
Department of Computer Science and Technology
Heilongjiang University
Harbin, China
[1]wangjinyong818@163.com, [2]hxw@hlju.edu.cn

Abstract *- We propose a simple approach for Chinese syntactic parsing postprocess in this paper. It uses verb subcategorization syntactic mode to match n-best candidate parsing trees outputed from baseline parser system. We extract various features of verb subcategorization from train corpora. And use those features of verb subcategorization extracted from train corpus to rerank the n-best list via a similar pattern matching approach, and with the rule-based method, but no use statistic information. We called this method as rule-based reranking. The result shows our approach reaches a good performance.*

Keywords: Rule-based Chinese Syntactic Parsing Postprocess; Verb Subcategorization

1 Introduction

Recently there has been a wealth of research into the syntactic parsing based on corpus. Syntactic parsing is the task of identifying the phrase or clause structure of sentence in natural language process. Moreover, reranking has proved to be a successful strategy for syntactic parsing postprocess. In this paper, we focus on parsing the part[1] of Tsinghua Chinese Treebank(TCT)[1]. Over the last few decades, with the field of syntactic parsing, a gradual but marked shift in the focus of language research has taken place. Besides parsing reranking for postprocess [2,3,4,5,6,7] have been proposed and promising performance improvements have been reported. In addition, combination methods for postprocess[8,9,10] has also been shown to be another effective technique to improve parsing performance. Now we present a brief survey of the reranking task. Reranking is technology that employs a rich set of local and global linguistic features [7] to rerank the n-best output on the forest level or tree level. It has been successfully applied to some NLP problems, especially to the problem of parse reranking. For example, Ratnaparkhi [11] noticed that the F1-Score could be improved to 93% from 87%, if the oracle parse was successfully detected by ranking the 20-best parsing results generated by his maximal entropy parser. Charniak [3] reranked the n-best parses by reestimating a language model on a large number of features.

Although reranking[2,3,4,5,6,7] has extremely improved the performance of syntactic parsing, they is inadequate using a verb feature from the reranking model. In this paper, we proprse a kind of verb subcategorization-based syntactic mode match approach to rerank n-best outputs of individual parsers, and adequately use rich global feature, verb-subcategorization in the sentence to improve the performance of syntactic parsting. For simplicity we use two popular generative statistical parsing models: Stanford Parser [12, 13], factored model, and Berkeley Parser [14,15], unlexical model. We follow conventional informal usage and refer to these as the "Stanford" and the "Berkeley" parsers respectively. Firstly, we applies Stanford Parser and Berkeley Parser to output n-best parse trees respectively, then extracted verb subcategorization syntactic mode feature from train data set, as shown figure 1 below. We are interested to see whether the same kinds of features improve the performance of both the Stanford and the Berkeley Parsers, or whether successful reranking needs particular feature that are specially tuned to the parser that it is deal with. Finally, The performance of the reranking trained on the union n-best parse trees is focused by us. As mentioned above, combination approach is other more complex ways that has been previously demonstrated to improve overall accuracy, whether we use the relatively simple method used here improves parsing accuracy as well.

We firstly perform experiments using verb subcategorization-based rule syntactic parsing postprocess for Chinese language on the Tsinghua Chinese Treebank(TCT). It is hoped that this brief and necessarily over simplified investigation will encourage researcher about verb subcategorization to take them into practice. Compared to nouns, adjectives and other parts of speech, verb has the most complex form and the strongest activity in the syntactic structure. Most of the other elements of sentence must be to some extent dominated by the verb, moreover verb is usually the most elements in the sentences as a predicate or predicate head. It is the most abundant. Therefore, it is a significant that researches on verb subcategorization in the sentence.

We perform relative experiments on the Tsinghua Chinese Treebank (TCT). Experimental results show that our final results, an F1-Score of 88.67 on Chinese, close the previously best reported systems reported F1-Score of 88.77 in the CIPS ParsEval-2009. This convincingly demonstrates

[1]The part of Tsinghua Chinese Treebank is the test corpus using in the CIPS ParsEval-2009.

the effectiveness of our proposed verb subcategorization-based rule syntactic parsing postprocess for Chinese language. Our study also shows that the pattern matching approach is more effective and simple.

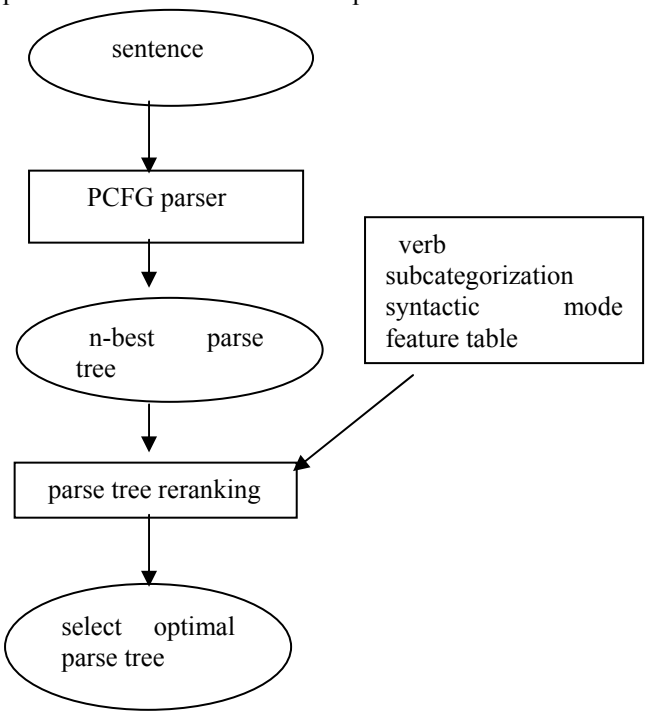


Figure 1. the Flow Chart of Syntactic Parsing postprocess

The paper is organized as follows. In Section 2, we summarize the previous works on combination and reranking. We introduce the theory of verb subcategorization syntactic mode match approach in section 3. In Section 4, and discusses our method. in Section 5, we report our experimental results and then discussion and conclusions in Section 6.

2 Relation work

As discussed in the above section, parsing trees combination and reranking are two basic methods to improve parsing performance by postprocessing parsers' n-best outputs.

Regarding the parser reranking, Collins[2] proposes reranking method to improve syntactic parsing performance. Firstly, apply a first model for a sentence to generate n-best syntactic parsing trees, and then apply the second model to rerank n-best syntactic parsing trees , from these results, select out an optimal parsing tree as the final result. Collins [2] published results (89.9% precision and recall rate of 89.6%) is recognized as the year's best English language syntactic parsing results. Charniak and Johnson [3] proposed a model based on maximum entropy discriminative resorting method. In the syntactic parsing trees of 50, select a candidate and the best syntactic parsing tree as the final syntactic parsing result. F-Score of this method reached 91.0%, is the best English parsing result achieved. Huang [7] utilizing forest reranking method that reranks the packed forest of exponentially many syntactic parse trees. His final result, an F1-Score of 91.7, outperforms both 50-best and 100-best reranking baselines, and is better than any previously reported systems trained on the English Penn Treebank.

Regarding the system combination study, Henderson and Brill [8] proposed two parser constituents combination methods, it breaks each parse tree into constituents, calculates the count of each constituent, then applies the majority voting to decide which constituent would appear in its final tree and get an F1-Score of 90.6. Sagae and Lavie [9] improve this approach by presented a reparsing scheme that produces results with accuracy higher than the best individual parsers available by combining their results. They report that F1-Score reached a score of 91.0. Fossum and Knight [16] provide parse hybridization, computing expected F1-Score using Minimum Bayes Risk parse selection and n-best outputs and reach a good test result in Penn Treebank. Of course, combination methods, also help to improve the performance of a highly accurate parser. Whereas about Chinese Syntactic parsing postprocess, Chen and Huang [17]using constitutes of n-best parse trees from the multiple parser to recombine the parse trees. They achieved the first result at the closing test in the Tsinghua Chinese corpus, F1-Score is 88.77 without head match.

However, the above-mentioned method of the postprocess of syntactic parsing included combination and reranking is almost about English corpus seems to be complication. In this paper, we present a simple approach to Chinese syntactic parsing using verb subcategorization-based rule pattern matching method to rerank the output of a baseline parsing system. This method can be seen in the following sections .

3 The theory of verbsubcategorization syntactic mode match approach

Suppose there is d class corpus and extracted subcategorization model β1, β2, β3, ... βd, each subcategorization model class corresponds to a grammar G generating the optimal parse tree T1, T2, T3, ... Td. So if a sentence can be generated Tj of grammar G, that is the sentence belongs to class βj. Input a sentence X, syntactic analysis in the treatment of subcategorization pattern recognition task is to determine which X belongs to a class, as shown Figure 2 .

For example, we gave a optimal tree sequence from train corpus, T= {t1, t2, t3, …, tn}, and extractedverb subcategorization syntactic mode in the same time, VSF={v1,v2,v3 …,vn}, then v1->t1, v2->t2, v3->t3, vn->tn, established one mapping between them. We say optimal tree can generate verb subcategorization syntactic mode, that is T->VSF. Instead, if input a sentence to generate n-best parse trees from test corpus, and extracted verb subcategorization syntactic mode, that is VSF*={v11,v12,v13, …,v1n}, when one in VSF*

from test data matched one in VSF from train data, we think VSF* generated the optimal tree, that is VSF*->T.

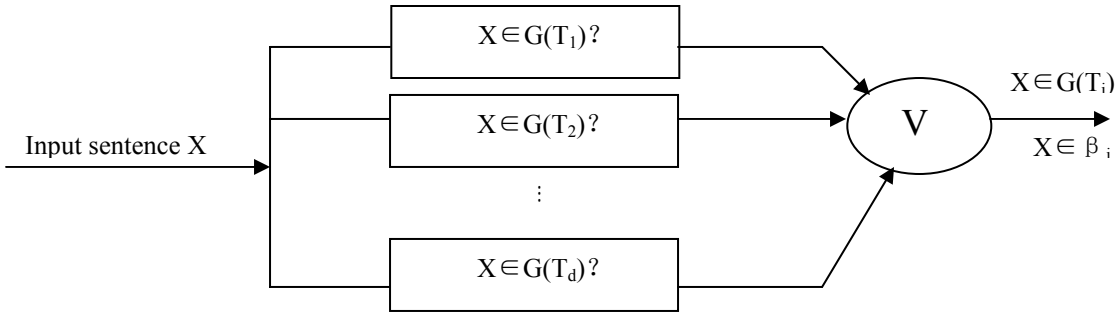


Figure 2. Theory of Verb-subcategorization Pattern Matching

From the above example, it could fully explain that the theory of verb subcategorization syntactic mode match approach is correct and simply effective. Below we will discuss the application of it in this paper.

4 Experiment setup

4.1 The adaptive parser and the corpus of TCT

The dataset we used is from the Tsinghua Chinese Treebank(TCT) that is adopted by Chinese Information ProcessingSociety of China for ParsEva-2009 evaluation.

We use Chinese syntactic parsing section in Tsinghua Chinese Treebank (TCT) is based event description units,The annotation in the dataset is different to the other frequently used Chinese treebank (i.e. CPTB) which combines both the constituent tree structure and the head information of each constituent. A total of 171 train data file in which the document, including BAIKE (23 articles) NEWS (125), HYL (23 papers); event description unit has 67172; test corpus a total 43 files; there are 16210 clauses released testing data. For simplifying syntactic parsing course, we neglected in one word[2] (origianl sentence was 16210, after ignoring one word, the sentence number 7839). In actual test, one word[2] is notation such as comma and full stop et al. Test sentences were entered with part of speech tagging, and we conduct exercises with unhead information corpus.

The Stanford Parser is based on the factored model with separate PCFG phrase structure and lexical dependence experts, whose preferences are combined by efficient exact inference, using an A* algorithm. and it is a lexicalized probabilistic parser that implements a factored product model. Berkeley Parser is based on unlexicalized parsing model. It uses an entirely data-driven approached to estimate the probabilities involved in the context-free grammars with EM training, usually beginning with the barest possible initial structure and uses a hierarchical coarse-to-fine scheme to refine the grammars, until the predicted syntactic structures fit the training data well enough to a certain degree. Moreover, it is a PCFG parser, written in Java, that can be trained on standardized collection of syntactically annotated data such as the Penn Treebank. Although both parsers are claimed to be multilingual parsers, in fact only accept training data in Penn Treebank format. To use these parsers to the Tsinghua Chinese Treebank (TCT) used in CIPS ParsEval-2009 task 5, we firstly transform the TCT training data into Penn Treebank format. Then, some slight modifications have been done, e. g, add ROOT support based on Penn Treebank parsing configuration into TCT sentences since the TCT roots can be –dj, -vp or –np and so on. For example, the following format transformed.

(1) The syntactic parsing tree format in Tsinghua Chinese Treebank (TCT) before conversion.
[dj 汉族/nR [dj 人口/n [ap 最/dD 多/a]]]
[dj [np 聚居/vN 地域/n] [ap 最/dD 广/a]]
[dj [dj 文字/n 产生/v] [ap 最/dD 早/a]]

(2) The syntactic parsing tree format in Tsinghua Chinese Treebank (TCT) after transforming Penn Treebank format[3].
(ROOT (dj (nR 汉族/nR) (dj (n 人口/n) (ap (dD 最/dD) (a 多/a)))))
(ROOT (dj (np (vN 聚居/vN) (n 地域/n)) (ap (dD 最/dD) (a 广/a))))
(ROOT (dj (dj (n 文字/n) (v 产生/v)) (ap (dD 最/dD) (a 早/a))))

4.2 Extract Verb Subcategorization Syntactic Mode Feature Table from Train Data set.

We extract verb subcategorization syntactic mode feature table from 22835 clauses in the train data, developing data is 9787 clauses, the ration of training and developing sets is 7:3,

[2]One word imply the no meaning word ,such as comma, period et al. TCT is different from Penn Treebank. TCT is simply syntactic parsing with Event-based split, namely, it is not full parsing and is similar chunk parsing.

[3]For the convenience evaluation of TCT, we use the word adding POS in the Penn Treebank format. This didn't influence syntactic test result, otherwise increasing test result effectively.

of course, they removed one word in the original 67172 clauses of Tsinghua Chinese Treebank (TCT). 7939 sentences were used for the test corpus, average 10.6 word/clause, that were not used in any way to train the system.

Transformation-based error-driven learning approach is EricBrill [18], proposed an effective learning approach. Figure 3 illustrates how transformation-based error-driven learning works. First, unparsing text is performing syntactic parsing, produced n-best output from parser. In syntactic parsing, various parser have been used, including Stanford Parser and Berkeley Parser. At the same time, original verb subcategorization syntactic mode feature table was extracted from train corpus. Then 1-best syntactic parsing output, that is parsed text was produced by postprocess step, then it compared to answer text which a manually annotated syntactic corpus. It is worth noting that the object function for comparing the corpus to the answer text and choosing transformation state is a F1-Score.When no transformation state can be found whose role reduces error beyond the object function(F1-Score), learning stops. New VSF will be finally got from train data set.

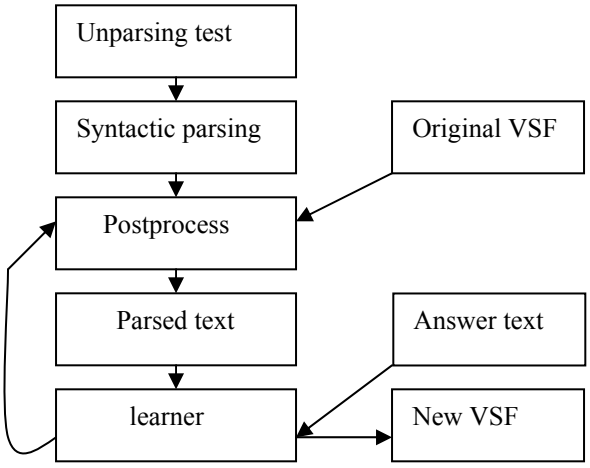


Figure 3.Transformation-based Error-driven Learning

Figure 4 shows an example of learning transformation. In this example, we assume there are four possible transformation states, t1 though t4, and that the objective function is the F1-Score. The unparsing developing text is processed by the postprocess step, and this results in parsed text is F1=0.6, determined by comparing the output of parsed text with answer text. Next, we apply each of the possible transformation states in turn and score the result of F1-Score. In this example, t2 result is the largest F1-Score, so t2 is learned as the first transformation. Following this, transformation t3 result is the largest F1-Score, so it is learned as a second transformation. After t2 and t3, no further highly F1-Score can be obtained from applying any of the transformation states, so learning stop.

We use the constant template[4] in the term of transformation-based error-driven learning course. Because of exhausted extracting VSF formal from train data set, we extract new VSF only need several run of learning and F1-Score convergence.

4.3 Postprocessing using Verb Subcategorization Syntactic Mode Feature Table

It is time to product 1-best output of syntactic parsing trees for the postprocess following the verb subcategorization syntactic mode feature table extracted from the train corpus with the transformation-based error-driven learning approach. In this research, pattern match approach were employed for the postprocess. The process is as follows, firstly, both Stanford Parser and Berkeley Parser were used to generate n-best output with negative probability for test data sentences respectively, and then we sorted them with the probability in descending order. Then, Using the feature of verb subcategorization in the VSF table compared mode match of VSF with the VSF of each sentence extracted from the test data. We count every matching numbers, and choose the most matching parse tree as the optimal one. If they match no one in the VSF table extracted from the train data set, no deal, 1-best output according to the maximum probability of sentences. We think this case is relatively short, because the most VSF extracted from the test data set can match the one in the VSF table extracted from the train data set. In here, VSF indicates verb subcategorization syntactic mode feature.

5 Results

We used the Tsinghua Treebank that is performing in closed test, and absolutely no other data or information could be used beyond that in the training data set.

For evaluating the test of postprocess from the n-best outputs of Stanford and Berkeley Parsers, the usual precision, recall and F1-Score are used, that is:

$$\text{Precision} = \frac{\text{num. of correctly recognized constituents}}{\text{num. of recognized constituents}}$$

$$\text{Recall} = \frac{\text{num. of correctly recognized constituents}}{\text{num. of total constituents}}$$

$$F1 = 2 * \text{Precision} * \text{Recall} / (\text{Precision} + \text{Recall})$$

Firstly, we conducted experiments with the Stanford Parser described section 4 using syntactic mode match method of verb subcategorization for the postprocess of syntactic parses. Table 1 shows the result. From it, we can see our method that took the 1-best outputing as based outputing result in all test cases with the ParsEval-2009 evaluation metrics. Using the entire Chinese test section of the Tsinghua treebank, our method using 10-best outputing subtly improves the performance by 0.01 (78.15-78.14) point. Using 50-best outputing for the postprocess, our method

[4]Constant template is a VSF table from train data set, we use to extract the new VSF table with learning approach from developed data set.

apparently improves the performance by 4.56 (82.70-78.14) point in F1-Score, representing 20.8% error rate reduction. These improvements convincingly demonstrate the effectiveness of our method based on rule syntactic parsing postprocess.

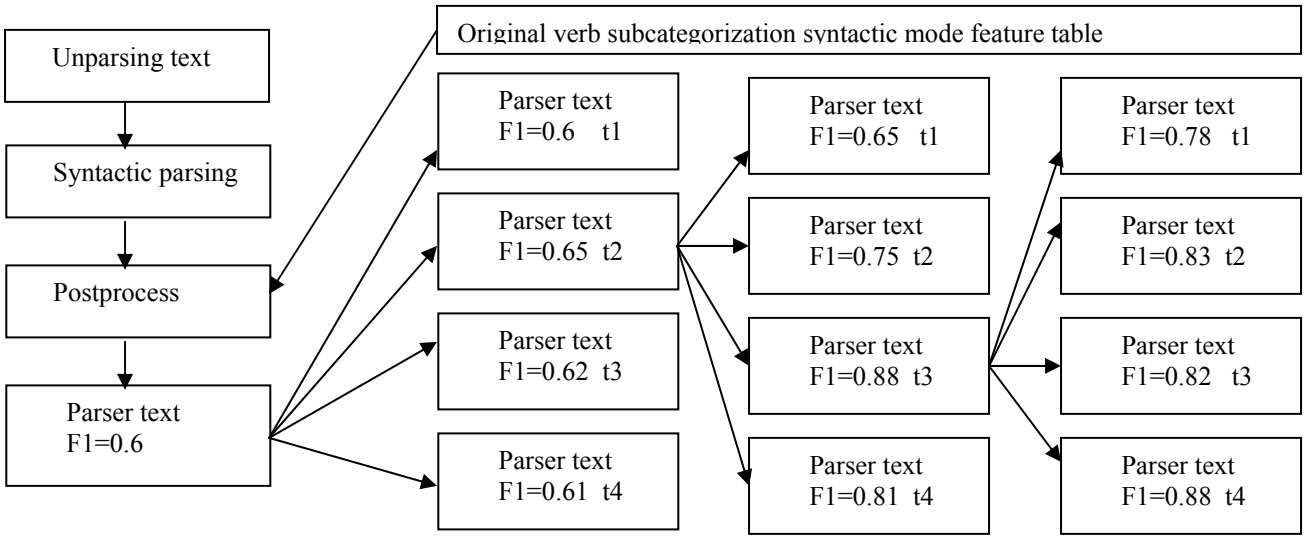


Figure 4. An Example of Verb Subcategorization Feature Using Transformation-Based Error-Driven Learning

Stanford Parser(%)	Precision	Recall	F1
1-best	78.44	77.85	78.14
10-best	78.40	77.90	78.15
50-best	82.36	83.05	82.70

Table 1. The Performance of Stanford Parser

Secondly, experiments were conducted using above using the same method. Table 2 shows the result below. from it, we can see 1-best output from Berkeley Parser is much higher than 1-best output from Stanford Parser. This is because Berkeley Parser is an implementation of the unlexicalized parsing model, which is more generalized and almost language-undependent, while Stanford Parser is based on lexicalized model and factored model. Moreover, Chinese language is different from English language, so lexicalized Stanford Parser didn't generate effectively lexicalized grammar for Chinese syntactic parsing. Our method using 10-best outputing subtly improves the performance by 0.55 (86.57-86.02) point in F1-Score, Using 50-best outputing for the postprocess, our method apparently improves the performance by 1.97 (87.99-86.02) point in F1-Score, representing 14.1% error rate reduction.

Berkeley Parser(%)	Precision	Recall	F1
1-best	86.07	85.97	86.02
10-best	86.61	86.54	86.57
50-best	88.07	87.91	87.99

Table 2. The Performance of Berkeley Parser

Finally, experiment were conducted using the method that combining Stanford and Berkeley Parser 10-best and 50-best output respectively, in doing exercise, used 10-best output from Berkeley Parser as based output, while 10-best output from Stanford Parser were added to it. Of course, in turn it is true. Removing duplicate elements in 10-best output from Berkeley and Stanford Parser. 50-best output is the same approach as well as 10-best output. From the following Table 3, current best indicates the best performance of parser evaluated on official testing set by the official evaluation program in the CIPS ParsEval-2009. But its Precision and Recall did not record. Experimental results of 10-best output present that our F1-Scores of 87.01. Moreover 50-best output, F1-Score reached 88.67, is very closely state-of-the-art F1-Score 88.77 in the syntactic parsing evaluation of the CIPS ParsEval-2009.

Stanford +Berkeley (%)	Precision	Recall	F1
current best			88.77
[5]20-best	87.05	86.97	87.01
[6]100-best	88.65	88.69	88.67

Table 3. The Combining Performance of Stanford and Berkeley Parsers

6 Conclusions

In this paper, we propose a similar matching method of verb subcategorization-based rule syntactic parsing postprocess for Chinese language. Compared with previous

[5]20-best indicates the combination of Stanford and Berkeley 's 10-best outputting respectively.

[6]100-best indicates the combination of Stanford and Berkeley's 50-best outputting respectively.

methods, our method is able to use global features, including no using algorithm of the parsing tree probability calculated by the individual systems, whereas a similar pattern matching method numbers. We verify our method by combining the n-best parsing trees of two representative parsing models, lexicalized model and unlexicalized model, and outputing n-best syntactic trees respectively. Experimental results show our method is very effective and gets a good performance, close the state-of-the-art results reported in the CIPS ParsEval-2009 on Chinese syntactic parsing.

An advantage of using syntactic parsing postprocess is that numerous global features can be added robustly because rule-based rerank can learn to choose only the optimal ones.

But, disadvantage of using syntactic parsing postprocess is that some sentences have not verb subcategorization (account for 6%), we know VSF is the important role in the sentence, though they are quite short numbers, syntactic parsing also suffers in the case. In the future, we will explore more features of rule-based syntactic parsing and study the verbless subcategorization-based methods for syntactic parsing postprocess.

7 Acknowledgments

We would like to thank the Berkeley Parser and Stanford Parser providers , and this paper is based on National Science Foundation of China for its support of this research under grant number 60773069, 60873169 .

8 References

[1] Qiang Zhou. Annotation Scheme for Chinese Treebank. Journal of Chinese Information Processing, vol 18(4), pages 1-8. 2004.

[2] Michael Collins. Discriminative reranking for natural language parsing. ICML-00, pages 175-182. 2000.

[3] Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine-grained n-best parsing and discriminative reranking. ACL-05, Pages 173-180. 2005.

[4] Michael Collins and Terry Koo. Discriminative reranking for natural language parsing. Computational Linguistics, 31(1):25–70, 2005.

[5] Taku Kudo, Jun Suzuki and Hidiki Zsozaki. Boosting-based parse reranking with subtree features, Annual Meeting of the Association of Computational Linguistics (ACL), pp.189-196, 2005.

[6] Markus Dreyer, David A. Smith and Noah A. Smith Vine parsing and minimum risk reranking for speed and precision. In *Proceedings of the CoNLL Shared Task*, pages 201-205, 2006.

[7] Liang Huang. *Forest Reranking: Discriminative Parsing with Non-Local Features.* ACL-HLT-08, pages 586-594. 2008.

[8] John Henderson and Eric Brill. *Exploiting diversity in natural language processing: combining parsers.* EMNLP-99. 1999.

[9] Kenji Sagae and Alon Lavie. Parser combination by reparsing. In Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers, pages 129–132, New York City, USA, June. Association for Computational Linguistics. 2006.

[10] Daniel Zeman and Zdeněk Žabokrtský. *Improving Parsing Accuracy by Combining Diverse Dependency Parsers.* IWPT-05. 2005.

[11] A. Ratnaparkhi. Learning to Parse Natural Language with Maximum Entropy Models. Machine Learning, 1999, 34(1-3): 151-175.

[12] Dan Klein and Christopher Manning. Fast exact inference with a factored model for natural language parsing. In Advances in Neural Information Processing Systems 15 (NIPS 2002), pages 3–10. MIT Press. 2003a.

[13] Dan Klein and Christopher D. Manning. 2003b. Accurate unlexicalized parsing. In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, pages 423–430, Sapporo, Japan, July. Association for Computational Linguistics. 2003b.

[14] Slav Petrov and Dan Klein. Improved inference for unlexicalized parsing. In Human Language Technologies The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference, pages 404–411, Rochester, New York, April. Association for Computational Linguistics. 2007.

[15] Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. Learning accurate, compact, and interpretable tree annotation. In Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, 2006: 433–440.

[16] Victoria Fossum and Kevin Knight. Combining constituent parsers. In Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers, pages 253–256, Boulder, Colorado, June. Association for Computational Linguistics. 2009.

[17] Xiao Chen，Changning Huang，Mu Li and Chunyu Kit. Better Parser Combination. JSIP-2009. 2009.

[18] Brill, Eric. Transformation-based error-driven learning and natural language processing: a case study in part of speech tagging. Computational Linguistics, 21(4):543--565. 1995.

SESSION

WORKSHOP ON INTELLIGENT LINGUISTIC TECHNOLOGIES, ILINTEC'11

Chair(s)

Dr. Elena B. Kozerenko

Graph Decomposition and Its Use for Ontology Verification and Semantic Representation

Julia M. Taylor
CERIAS & Linguistics
Purdue University
West Lafayette, IN, USA
jtaylor1@purdue.edu

Victor Raskin
Linguistics & CERIAS
Purdue University
West Lafayette, IN, USA
vraskin@purdue.edu

***Abstract*—This paper explores the use of a graph decomposition technique applied to an ontology, designed as a basis for understanding natural language text. It demonstrates how the technique can help with ontology verification, detecting common human errors in concept acquisition, as well as having some ramifications for semantic processing of multiple noun expressions, among other natural language entities. Ontology verification with the help of graph decomposition is illustrated on the phenomena of inverse properties, transitive properties, and hierarchical (ancestor/descendant) properties. The ontology on which the technique has been tested is that of Ontological Semantic Technology, a significantly modified version of Ontological Semantics [5].**

***Keywords*—ontological semantic technology, ontology verification, natural language, graph decomposition**

I. Exploiting the Formal Side of Ontological Semantics

Ontological Semantics ([8] and references there) has been motivated primarily by representing natural language meaning comprehensively and deeply rather than not at all or selectively and shallowly. It separates its resources into a language independent property-rich ontology, which reflects the knowledge of the world, and language-dependent lexicons where every sense of every word or phrasal is defined in ontological terms and the knowledge of a specific language is accommodated. At the same time, the approach has used a mathematical foundation and a reasonably clear but loosely defined mechanism (see, however, [8]:7.1.1, 7.1.6) that has been taken for granted but not protected from mis- and abuse in acquisition. Its ontology was definitely more engineering, in the sense of being result-oriented, rather than formal, i. e., fully dedicated to being built in strict accordance with the mathematical rules—and that prevented the school from taking conceptual and processing advantage of its own presumed mathematical properties. The formal ventures within the Ontological Semantics of the 1990s were rare, and even when incorporated into its main body ([8]:8.3.1-8.3.2), they were not implemented to any significant extent. Just about the only mathematical property acted upon was the hierarchical IS-A property on which simple inheritance is based, but even there, the not so rare phenomenon of multiple parenting was never accommodated sufficiently clearly and consistently, and the availability of a reliable acquisition tool that would preclude human inheritance errors was sporadic. Besides, not to quote Bill Clinton, "what is is" was never clearly defined (cf. the classical [1] and its follow-ups). As a result, the so-called legacy resources [7], resulting from the research efforts and small academic proof-of-concept implementations of the 1990s, remain quite faulty, both formally and, accordingly, substantively.

The Ontological Semantic Technology (OST) of the late 2000s [9,5,10], besides modifying and, on occasion, departing from the earlier tenets, has been also increasingly more conscious of its formal foundations and interested in deepening their understanding and significance for adequate meaning representation and processing, such as inferencing and reasoning, where important research outside of the approach has been done on their logical aspects, and while not deep or complex enough for real NL processing needs, it would be unavailable for use by OST unless the latter strictly conformed to its own professed formal properties as well as developing and improving them to its substantive advantage. One major step in this direction was virtual ontology [10] that suggested a totally new perspective on the bifurcation of Ontological Semantic resources into the ontology and lexicons.

This paper extends the mathematical foundations of the OST ontology by introducing—or rather adapting for the purpose—the notions of (de)composition and derivation. Seen as a graph (see [4] for the basic background), the ontology is an incredibly complex and interweaving aggregate of vertices and edges. Decomposition slices this graph into multiple graphs, each organized around—or derived for—one property. The paper also shows that decomposition helps to verify certain features of the ontology, detecting and removing human error with regard to these features. It can be used as well for processing multi-noun expressions, or noun chains, where the nouns in question do not provide a hint to the properties that connect them. Graph decomposition helps here both directly and indirectly: the former is achieved by finding the property whose decomposed graph yields the shortest path between the two concepts underlying the constituent nouns; the latter is provided through ontology verification, thus improving the direct decomposition results. The accurate processing of noun chains may come from the background information contained

This research was supported in part by the National Science Foundation Grant No. 1012208 (Division of Computer and Network Systems, Trustworthy Computer Program

in the ontology or gathered from elsewhere in and outside of text, and this paper will rely entirely on the ontological information rather than on the linguistic or extralinguistic context, thus providing a predominantly paradigmatic rather than syntagmatic solution.

In general, ontology verification is a crucial process, and it has been approached, mostly in formal ontology, for over a decade. A major step forward was OntoClean [2,3,12], a comprehensive system for checking on the intensions of the central principles and types of relations in formal ontology, the major premise being that any deviation from the prescribed rigor would result in crippling the engineering applications of ontology. Realizing that OntoClean is rather rarefied to use practically—and, in fact, has not seen too many practical applications—[11] proposed a somewhat more accessible application model. AEON, for using OntoClean, weaving some elements of natural language analysis into it. Closer to our own enterprise, [13] suggested an experimental model for evaluation an ontology on the basis of its performance in an application. This is indeed our ultimate stance on ontology verification and evaluation. We do, however, believe that quality control is essential in the process of ontology acquisition, long before an application is ready to work and expose errors that should be corrected. Our approach is different also because our ontology parsimoniously supports a set of large lexicons and models deep, comprehensive meaning rather than the shallow semantics that the ubiquitous [6] ascribes to ontologies in an attempt to endear them to the proponents of statistical and machine learning methods. Additionally and most importantly, OntoClean and its spin-offs are concerned with taxonomy, or vertical cleaning. We assume such cleaning to have been accomplished and focus instead on horizontal cleaning, that is the properties other than is-a.

Most processing errors in OST and other rule-based NLP systems are due to acquisition errors, especially in the ontology, where a mistake in making a restriction too broad or too narrow, for instance, has serious ramifications, as opposed to a lexicon error that affects only one entry. In the following sections, we demonstrate how graph decomposition works and how it can help with ontology verification and what properties lend themselves to such verification most readily.

II. Graph Decomposition

We treat ontology as a lattice that contains concepts and relationships among them. Structurally, an ontology can be considered to be a directed graph, where the relationships serve as edges. We can then exploit the structure and properties of a graph to verify the intuitive structure of an ontology, constructed from the linguistic point of view. Unlike a typical graph, where the edge has only one meaning per graph, and that is that of a (directional) path, cost, or other single-meaning metric, most edges in the ontology have different meanings, namely those of a property that represents a relationship between the connecting conceptual vertices. Looking at an ontology through a one-property prism can provide valuable insight, as we hope to show in this paper.

Prior to decomposing the ontology, the inherited information of each concept has to be taken into account. A lot of concepts inherit information from their ancestors. A concept may either inherit a property and a filler from its parent, or it may narrow down the filler of its parent using the same property, or it may add a new property that a parent concept does not have. The inherited information can be explicitly restated in the child concepts, for the simplicity of calculation, or be adjusted by the weight depending on the distance from the source of inheritance. Whenever properties themselves are arranged in a hierarchical order, a parent property used in a concept can be legally narrowed down to the child property with the same or narrower filler.

We look at an ontology as a composition of graphs, one graph per property, where each graph has a single meaning for its edges. For example, the IS-A graph contains a hierarchy of the concepts; the AGENT graph only connects vertices that are in the AGENT relationship with each other.

The ontology is, then, an ordered pair $O = (C, P)$ with a set C of concepts together with a set P of properties, which could be looked at as a two-element ordered subset of C, as each property connects two concepts. In other words, our ontology is a labeled multigraph, formally defined as:

(1) $O = (\Sigma_C, \Sigma_P, C, P, s, t, l_C, l_P)$, where

- C is a set of concepts or nodes
- P is a set of properties or multiset of arcs or edges
- Σ_C is a finite alphabet of the available concept labels
- Σ_P is a finite alphabet of the available property labels
- $s: C \rightarrow P$ is a map indicating the source node of an arc
- $t: C \rightarrow P$ is a map indicating the target node of an arc
- $l_C: C \rightarrow \Sigma_C$ is a map describing the labeling of conceptual nodes
- $l_P: P \rightarrow \Sigma_P$ is a map describing the labeling of property arcs

When a weighted ontology is considered, $O = (C, P, w)$, with nonnegative property weights $wst \geq 0, (s, t) \in C$ instead of $O = (C, P)$.

A. The structure of properties and their effect on decomposition

There are two cases that should be looked at in ontology decomposition: when properties are independent and unordered and when they are arranged in a partial order.

In the first case, when the properties are unordered, the ontology can be decomposed such that:

(2) $O=\{O_1, \ldots O_i, \ldots, O_n\}$, where $O_i = (\Sigma_C, p_i, C, P_i, s, t, l_C, l_P)$, $p_i \in \Sigma_P$ and $||P||=n$, with i indicating each property.

In other words, for every property, a sub-ontology O_i can be created that spans some concepts in the ontology using only the chosen property.

The situation is more difficult in the second case when the properties are ordered, especially as a hierarchy. For example, we may have a property LOCATION with 3 children: PATH, START-LOCATION, END-LOCATION, and we are constructing a LOCATION graph. The question arises: should its children be included on its graph? If we follow (2), then none of the concepts connected through PATH, START-LOCATION, END-LOCATION will be shown here, but intuitively, they should be, possibly with a slightly different weighting preference.

We then propose the following construction of any O_{iC} such that P_i is a parent of P_j:

(3) $O_{iC}=O_i \otimes O_j$ where l_P of O_j takes preference over l_p of O_i.

It is possible to construct a weighted graph instead, where arcs corresponding to O_j and O_i are included, but with different weights, which are determined at the ontology verification stage:

(4) $O_{iC}=O_i \otimes O_j$ where both l_P of O_j and l_p of O_i are included with some weight.

Trivially, then,

(5) if P_i is a parent of P_j, then $O_i = O_j \otimes X$, where X is some graph and O_i is not O.

It follows from (5) that if O_k is a subgraph of O_i, then P_k is a child of P_i, and should be seen as such in the ontology O.

B. *What type of graph are we looking based on source and target*

O_i can be further described based on the relationships between all pairs of (s, t) with respect to a chosen arc set. In many cases, ontological properties have non-intersecting domains and ranges, thus forming two disjoint sets of S and T, with O_i resulting in a bipartite graph. This situation is especially typical for properties that connect EVENTS and OBJECTS, such as AGENT, BENEFICIARY, etc, and their inverses AGENT-OF, BENEFICIARY-OF, etc.

On the opposite side of the spectrum, are decomposed graphs whose domains and ranges are the same set. It is especially interesting to look at these graphs for transitive properties for ontology verification purposes, which is one of the topics in the next section.

III. Applications of Graph Decomposition: Ontology Verification

In this section, we will illustrate, on a number of property type examples, how graph decomposition may be used to detect acquisition errors. We will briefly look at the cases of inverse properties, transitive properties, and cases where properties are parents with children. We will see how an acquirer can do something very wrong while not committing any error from the point of view of the syntax of the formalism nor from the point of view of its semantics, in the particular case of the acquired material, but it will create a contradictory situation in that part of the ontology.

A. *Inverses*

What is of interest in graph decomposition, from the ontology verification standpoint, is that some errors in inverses can now be detected by looking at the composition of O_i and O_j, where the former is the "direct-property' graph and the latter its inverse, for instance, INSTRUMENT and INSTRUMENT-OF. Figure 1 shows O_i and Figure 2 the composition graph of O_i and O_j.

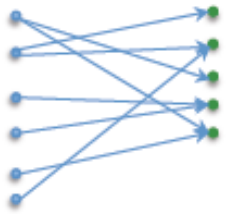


Figure 1. Property graph, where blue vertices indicate concepts, green vertices indicate property fillers

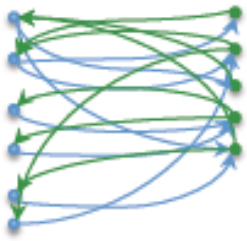


Figure 2. A property (blue edges) and its inverse (green edges)

A concept like DIG will, of course, have a concept like SHOVEL as its instrument. Correspondingly, SHOVEL will be described as INSTRUMENT-OF for DIG. It may lead to a situation where DIG will be erroneously constructed from the inverse property as the only instrument of SHOVEL or, conversely, that it will be set up as the only concept which is INSTRUMENT-OF for SHOVEL. The decomposed graph on the INSTRUMENT-OF property will, however, show that there are other nodes connected to DIG with this property. It is this disparity that will detect the inverse acquisition errors of this type. While no syntactic errors in the formalism are made in this type of error, a semantic inconsistency is detected: it is true that SHOVEL is indeed INSTRUMENT-OF for DIG and, conversely, DIG does have SHOVEL as an (but not the) INSTRUMENT. The mistake is that there are not the only possible such fillers for their respective properties, and leaving the error uncorrected will lead—and has led—to serious processing mistakes.

B. *Transitive properties*

Let us assume that there is a transitive property P_i in the ontology, resulting in a transitivity chain from A to B to C to D to E, as shown on Figure 3.

Figure 3. Transitive property graph, chain of interest is in green

The property connecting these vertices may be, for instance MADE-OF, the inverse of MATERIAL, with *A* = NECKLACE, *B* = BEAD, *C* = GLASS, *D* = SAND, and *E* = SILICA. The acquisition error is between *A* and *B*: in frequently occurring cognitive slippage, the acquirer misunderstood the ontological property MADE-OF as an English phrasal, which may have a different sense, that of 'constructed of' rather than 'composed of.' The error will be hiding in the ARTIFACT branch and not immediately visible in the branch containing natural materials. The graph decomposed on the MADE-OF property will present the whole path to an ontology engineer, who will respond negatively to the implied question whether necklaces are really made of sand or silica by cutting *A* from at least *D* and *E*, as per Figure 4.

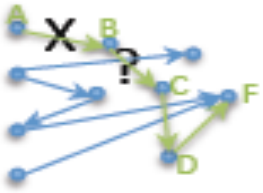


Figure 4. Transitive property with a mistake

C. Hierarchical properties

Some properties in the ontology are structured in a hierarchy, such as LOCATION, START-LOCATION, END-LOCATION, with LOCATION being the parent of the other two. Any concept that has a parent property can have child properties, but then the child property overrides the parent one. For example, TRANSFER-LOCATION, a descendent of EVENT, can have a START-LOCATION and END-LOCATION, but no longer a LOCATION which EVENT has. Such constraints on the domains can be trivially controlled. It is more difficult to control the range in such cases, especially when something can be in the range of a child property for one concept, and the range of a parent property for a different concept. For example, a CITY can be in the range of END-LOCATION of TRAVEL as well as in the range of LOCATION of OCCUPY. On the other hand, if we look at CITY, it is in the domain of END-LOCATION-OF (range TRAVEL) and LOCATION-OF (range OCCUPY) at the same time. If we go with the trivial domain validation for these properties, we would come up with an error, even though this is a perfectly legal situation. Moreover, CITY could have had LOCATION-OF any EVENT, which does have an error.

The weighted graph decomposition of the hierarchical properties can be used to create a simple solution: create a decomposition of O_i, where *i* is a parent property, according to (4). If there is a pair (*s, t*) such that they are connected by more than 1 arc, as shown in Figure 5, flag these nodes as erroneous and alert an ontology engineer.

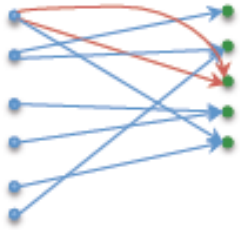


Figure 5. Hierarchical properties

Domain decomposition also makes it possible to check if one property could potentially be a parent of another property and alert an ontology engineer. A property *Pp* is considered to be a parent property of a Property *Pc*, if there exists O_{Pc} that forms a subgraph of O_{Pp}, in other words, every vertex of O_{Pc} is in O_{Pp} and every act of O_{Pc} is in O_{Pp}, as shown in Figure 6. If the ontology engineer decides that such properties were not meant to be in a parent-child relationship, at least one arc should be created to indicate that such relationship is incorrect.

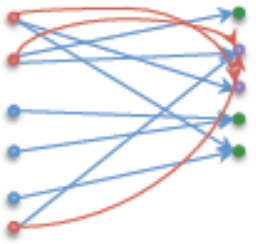


Figure 6. Domain decomposition of hierarchical properties

D. Practical applications of ontology verification

The results of ontology verification that lead to correction of ontological errors are best seen in the working of noun chains, i. e., *N* + *N**... The reason noun chains are an obvious demonstration of how important the correctness of the ontology is, is due to the fact that such chains do not explicitly provide a property that can connect the nouns. Such a property or several properties can only be found with the help of an ontology. Consider, for example, the noun chain *goat milk bottle*. No matter what the sentence says, the relationship between *milk* and *bottle* and *goat* and *milk* will not be specified there. There is also no preposition that can suggest the relationship. Thus, all the work is carried by the ontology and knowledge contained in it.

It is possible to interpret these noun chains with the help of graph decomposition. For each pair of concepts corresponding to the words, the shortest path is found in each O_i decomposition, thus resulting in a list of (value, property i) pairs. It is reasonable to suspect that the smallest non-zero

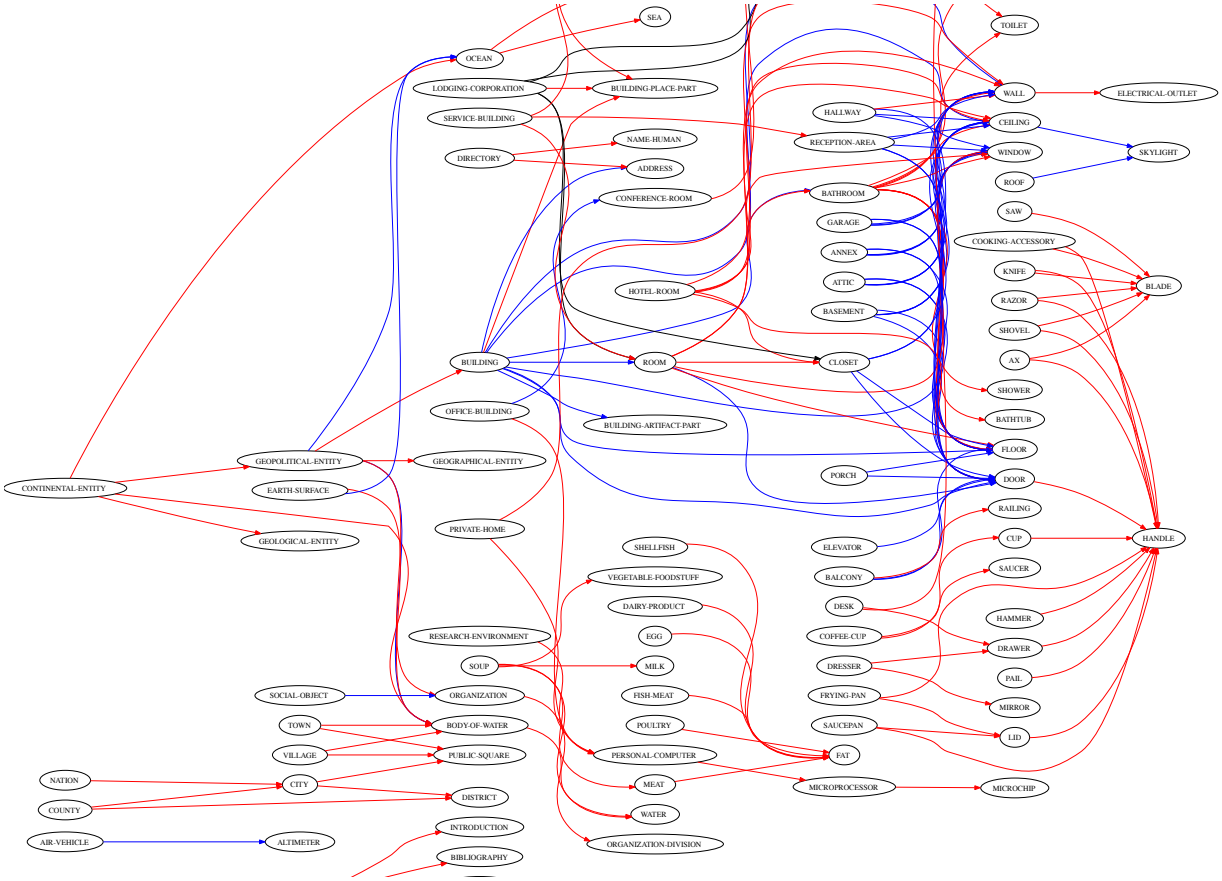


Figure 7. Fragment of HAS-OBJECT-AS-PART property graph

value points to the best property or a set of alternative best properties. For example, it is reasonable to suspect that (1, CONTAINS) will be among the best pairs for anchoring concepts of *bottle* and *milk*. Similarly, it is reasonable to assume that (1, ORIGIN) will be among the best pairs for anchoring concepts behind *goat* and *milk*, resulting in the following text meaning representation: (bottle(contains(milk(origin(goat))))).

Let us suppose that there is also a concept of ALCOHOLIC-DRINK, and it is CONTAINED-IN a BOTTLE. CONTAINED-IN is an inverse of CONTAINS. If the inverses were not done correctly and the mistake not caught, for instance, with the help of graph decomposition, as shown in the previous section, it is possible that ALCOHOLIC-DRINK would THEN override the concept LIQUID that a *bottle* can contain. Then, information in the ontology will be such that a *bottle* cannot contain *milk* (although *milk* is still CONTAINED-IN a *bottle*), which would result in the absence of CONTAINS from the list of properties with the shortest path between *milk* and *bottle* and thus prevent the semantic text analyzer from arriving at the text meaning representation above.

Similar mistakes can be avoided with the described verification of hierarchical properties in such examples as *Chicago trip* (with *Chicago*—assuming the city sense—being the END-LOCATION or DESTINATION of the trip, rather than a LOCATION). In this case, the list of path-property pairs would consist of only one selection, instead of two, thus removing the ambiguity that the analyzer will signal but that will not really be there.

IV. EMPIRICAL VERIFICATION

The question that we would like to address in this section is how useful graph decomposition turns out to be in a real life system. For the testing scenario, we are using a public domain ontology, accessible through [7] and commonly referred in the literature [3, 6, 8] as the MikroKosmos ontology. We purposefully selected an ontology that had not been touched for years, and the rank-and-file ontology engineers who were most familiar with the ontology had long moved on since. We are thus looking at a sufficiently unfamiliar knowledge resource.

The question that we want to address is whether we can estimate the lower bound of changes—i. e., the bare minimum of corrections—that would have to be made to this ontology for it to be useful for natural language applications that we have in mind. It should be noted that such measures are not necessarily an indication of the quality of the ontology, either way, but

rather of the feasible modifications for a particular application. It is assumed here that the hierarchical, or "sortal" correction, addressed in [2-3] is done separately and possibly previously.

In addition to the graph decomposition property verification techniques outlined in the section above, another trivial but very important feature should be mentioned. Once the decomposition is accomplished, it is highly convenient to look at the resulting graph to spot inconsistencies and errors practically "at a glance." For example, it took less than a minute for the authors both experienced knowledge engineers, who had not looked at this ontology for years if at all, to spot an mistake in 1 out of the 5 non-inherited usages of the property SUBJECT in the ontology; about 5 minutes for 27 mistakes in the 74 non-inherited usages of the property HAS-EVENT-AS-PART. It took about 10 minutes to spot several types of inconsistencies in 465 non-inherited usages of the property HAS-OBJECT-AS-PART. These particular inconsistencies were especially important to catch because they revealed that the ontology was not consistent with regard to mereological principles, thus compromising the integrity of the property, and thus ensuring incorrect interpretations, especially since the property is transitive.

The inverse properties were checked in a similar manner, according to the principles outlined in Section III.A above. In less than a minute, property SUBJECT and its inverse SUBJECT-MATTER were inspected and an inconsistency in 1 usage out of 11 was discovered. Fig. 7 above is the busiest section of HAS-OBJECT-AS-PART graph, with inverses marked in blue.

The error visibility of transitive properties on these graphs is quite impressive as well, making verification quite expedient. Thus, it took less than 10 minutes to examine the existing transitive links on the same HAS-OBJECT-AS-PART property. There were 31 errors spotted among the 278 links that connected more than 2 nodes.

V. Conclusion

In this paper, we explored the graph decomposition of an ontology and illustrated, on the example of the OST technology, how to use it for ontology verification with some ramifications for semantic representation. Ontology verification is crucial when humans and computers collaborate in a hybrid semi-automatic system of knowledge acquisition. This work can be seen as yet another step in increasing the degree of automation in the system for property-rich ontologies. It must be emphasized again that this is used in addition to, not instead of, standard logical inconsistency checks on the ontologies.

VI. Acknowledgements

As stated above, this research was supported in part by the National Science Foundation Grant No. 1012208 (Division of Computer and Network Systems, Trustworthy Computer Program). Some of the graphs are results of a software written for this grant by Mr. Blake Self, the resources for which have been maintained by Mr. Adam Hammer of CERIAS at Purdue.

References

[1] Brachman, R. 1983. What IS-A is and isn't. An Analysis of Taxonomic Links in Semantic Networks, *IEEE Computer* 16:10

[2] Guarino, N. and C. Welty. 2002. Evaluating Ontological Decisions with OntoClean. *Communications of the ACM.* **45**(2):61-65. New York:ACM Press

[3] Guarino, N., and C. Welty. 2002. Identity and Subsumption. In R. Green, C. A. Bean, & S. H. Myaeng (Eds.), *The Semantics of Relationships: An Interdisciplinary Perspective.* Pp. 111-125. Dordrecht: Kluwer.

[4] Harari, F. 1969. *Graph Theory*. Reading, MA: Addison–Wesley

[5] Hempelmann, C. F., J. M. Taylor, and V. Raskin 2010. Application-guided Ontological Engineering, *International Conference on Artificial Intelligence,* Las Vegas, NE, July

[6] Hovy, E.. 2005. Methodologies for the Reliable Construction of Ontological Knowledge. In: F. Dau, M.-L. Mugnier, and G. Stumme (eds), *Conceptual Structures: Common Semantics for Sharing Knowledge. Proceedings of the 13th Annual International Conference on Conceptual Structures (ICCS 2005)*, pp 91–106. *Springer Lecture Notes in AI* 3596..

[7] KBAE 2002. Knowledge Based Acquisition Editor. CERIAS, Purdue University, W. Lafayette, IN. http://kbae.cerias.purdue.edu:443, login: nlpgroup, password: ch@ng3m3

[8] Nirenburg, S., and V. Raskin 2004. *Ontological Semantics*. Cambridge, MA: MIT Press.

[9] Raskin, V., J. M. Taylor, and C. F. Hempelmann 2010. Guessing vs. Knowing: The Two Approaches to Semantics in Natural Language Processing, *Annual International Conference Dialogue 2010*, Moscow, Russia

[10] Taylor, J. M., V. Raskin, and C. F. Hempelmann 2010. On an Automatic Acquisition Toolbox for Ontologies and Lexicons in Ontological Semantics, *International Conference on Artificial Intelligence,* Las Vegas, NE, July.

[11] Völker, J., D. Vrandecic, and Y. Sure. 2005. Automatic Evaluation of Ontologies (AEON). In Y. Gil, E. Motta, V. R. Benjamins, M. A. Musen, eds. *Proceedings of the 4th International Semantic Web Conference (ISWC2005), volume 3729 of LNCS*, pp. 716-731. Springer Verlagerlin-Heidelberg, November.

[12] Welty, C. and N. Guarino. 2001. Support for Ontological Analysis of Taxonomic Relationships. *J. Data and Knowledge Engineering*. **39**(1):51-74. October,.

[13] Welty, C., R. Mahindru, and J. Chu-Carroll. 2004. Evaluating Ontology Cleaning. *Proceedings of AAAI-2004*. Pp. 311-314. San Jose:AAAI Press. July.

Textometry and Information Discovery: A New Approach to Mining Textual Data on the Web

E. MacMurray[1], **M. Leenhardt**[1,2],
[1]SYLED/CLA²T EA2290 UFR ILPGA Université Sorbonne Nouvelle Paris 3, France
[2]Le Semiopôle, Montreuil, France
[1] erin.macmurray@gmail.com, [2] marguerite.leenhardt@gmail.com

Abstract - *Most Text Mining tasks focus on local linguistic rules for detecting such elements as named entities, events and opinions: the goal here is to go beyond these local context boundaries by taking global dimensions into account. A robust method to mine textual data known as Textometry is not constrained by external resources and avoids problems such as the coverage limitations of standard dictionaries and at a higher level, domain-dependant resources. Textometry provides a new approach of exploring and comparing textual data. This paper studies the Textometric method and how it can be applied to the industrial context of mining named entities and their trends (opinions or events) in both French and American online news media: Le Monde and the New York Times. This paper focuses on bypassing certain costly steps in tasks related to mining information on Named Entities.*

Keywords: Textometry, quantitative linguistics, textual statistics, named entity mining, opinion mining

1 Introduction

It's no scoop that data- or the "quiet revolution" as Bollier [2] puts it- has grown tremendously since the availability of computing and databases, even more so since the dawn of the Internet. Data is not just conveniently stored in structured databases, it comes in the form of natural language: articles, blogs, forums are among some of the many formats in the mobile network for sharing information. This growing collection of content demands computer processing in order to dig or render visible information of interest. The detection and extraction of named entities in large compilations of text helps pin point potential zones of information corresponding to intense activity of the Named Entity in the media. In this paper we compare two intelligence application use cases where known statistical algorithms are applied as a method for mining information on named entities in online news articles from both Le Monde and the New York Times. Named Entities were used as an entry point for the analysis of the corpora. Then, statistical tools provided results for creating new Linguistic Resources (LR) in French and English for both the context of opinion analysis and event detection. This research puts forth a new approach to textual data analysis through methods for more industrial contexts, such as business and communication intelligence needs.

1.1 Mining techniques and natural language processing

Today, there are many natural language mining techniques: machine learning and information extraction through automatic semantic and morpho-syntactic patterns to name just a couple as discussed during the Message Understanding Conferences (MUC) [8]. Text Mining, generally seen as a subfield of Data Mining, is roughly defined as the processes used to extract and structure unstructured data [6]. Early work in text mining tried simply applying the algorithms developed for data mining without considering their specific unstructured nature [5],[9]. These applications showed how it was possible to use the methods of extraction sequences to identify new trends in a database [11]. However, textual data presents very different challenges from pre-structured data. Text Mining techniques often use a structuring phase of the information expressed in natural language in order to apply standard data mining strategies[6],[11]. The units of analysis used by these techniques rarely go beyond the sentence level and sometimes fail to consider their object of analysis, the text, as a component in and of itself. Here, our goal is to shift the focus from the sentence level to the text level by applying existing statistical strategies to discover patterns at this higher level in a corpus of unstructured textual data.

1.2 Data heterogeneity and web mining

Beyond the notion that textual data is unstructured, there is another major difficulty when putting in place a mining strategy dedicated to the qualitative analysis of textual content on the web- the heterogeneous nature of the data. As defined above textual data is unstructured information that simply keeps on expanding. Several factors must therefore be taken into account when developing a mining strategy. The first factor is the variety of physical media used to convey content: websites, social networks, forums, blogs, information portals. The structure of the content differs greatly depending on the medium. Although these media use meta-data in order to structure their online display and search possibilities, html/xml are only weak representations of the actual textual content. A second important factor in the process of producing content is the different writing strategies web-users exercise when exchanging on the web. They can, for example, generate a full text segment in writing an article or blog or simply intervene by leaving a comment on text already

produced. Thirdly, in the search process, different types of textual data must be considered: headline, column name, lead, article, date and time, legends and paragraphs, among others. These types of data will yield different results when being mined for information. These three factors (physical media, writing process, and text segments) make the task of pertinent information extraction complex, in other words, the development of robust systems plays a detrimental role in the management of these different variables. Moreover, in order to perform analyses, some structure will have to be given to the bulk of information gathered from these online media.

The goals for mining natural language are therefore twofold: (i) structuring free text for use by other computer applications, and (ii) providing strategies for following the trends and/or patterns expressed in the text.

We focus here on the latter, presenting the Textometric approach and showing the advantages of this method for text and information analysis issues. To this end, two subtasks of Information Discovery are considered: (i) mining Named Entities and (ii) gathering information for opinion detection and trend analysis.

1.3 Named Entity Mining for Opinion Detection and Information Discovery

Information Extraction systems have long attempted to group textual elements into Named Entities and relationships or template scenarios between these entities [8], [15]. Named Entity Recognition (NER) and Relation Templates continue to be hot topics today as they were during the MUCs, which can be noted by the number of open source technologies that have begun to undertake this task. The definitions attributed to what are called entities and relationships remain unsatisfactory. Entities are roughly defined as names of people, organizations, and geographic locations in a text [8]. They are perceived as rigid designators that reference 'real world' objects organized in an ontology [16]. However, these definitions fail to take into account the semantic complexity of named entities in terms of their surface polysemy and their underlying referentiality which aims at combining both the linguistic designation of an entity and the extra-linguistic level or the 'real world' object an entity refers to [16].

The situation is similar for Opinion Mining (OM). There is a terminologic instability resulting from the coexistence of *sentiment analysis* versus *opinion mining*, *evaluative stances* versus *opinion expressions*. The objects of OM and Sentiment Analysis are thus not based on consensus. Simply put, the technologies supporting sentiment analysis are related to classification tasks, whereas OM is derived from mining tasks. Named Entity Mining is deeply related to OM and Sentiment Analysis tasks. In following definitions given in [22] annotation objects, such as *agent annotation* or *target annotation,* rely on NER for their information discovery tasks in commercial applications. Although our method has yet to provide a satisfactory definition of named entities, combining both linguistic and computer science considerations, these objects remain vital access points for uncovering zones of information in the corpus.

2 A new approach to mining the web

As previously discussed here, information extraction techniques have often used semantic or content annotations to structure information of interest [6],[8],[11]. However, using qualitative coding- usually in the form of such morpho-syntactic or semantic annotations- to drive quantitative conclusions almost defeats the purpose of discovering unknown information in the text. Content annotations are not an abstraction of what is actually expressed in the text, but rather the vision of annotator creating them. This calls into question the accurate interpretation of results acquired using such basic information extraction techniques. Following MUC guidelines, precision and recall remain the gold standards for measuring such systems. However, "one man's noise is another man's data" [2], which clearly points out the difficulty in creating a generic system that can objectively process large quantities.

2.1 Textometric approach

Textometry is already well rooted in social science studies and quantitative linguistic research [10][11], mostly developed in France with numerous pioneers, Pierre Guiraud, Charles Muller, Jean-Paul Benzécri, Ludovic Lebart and André Salem. According to this approach, a text posseses its own internal structure that would be difficult to analyze by manual means alone. By applying statistical and probabilistic calculations directly to the textual units of comparable texts in a corpus [10][20] it becomes possible to analyze patterns and trends that would otherwise be obscured by the quantity of the textual units.

Information extraction techniques using qualitative coding can, therefore, be bypassed when studying textual data. Indeed, even basic preprocessing steps, such as lemmatization, can potentially hide distinctive features of textual units. Although, Textometry is not generally considered a text mining technique by the industrial community, because it is not fully automated, in following broader Text Mining definitions [6],[20], it seems an appropriate strategy for discovering related elements in a corpus when no predetermined information model is available. The Textometric analysis process relies on the interaction between an expert user and the system. The validity of the result interpretation is provided by and depends entirely on the expert.

2.2 Textometric objects

Textometry consists of seeing the document through a prism of numbers and figures, producing information on the frequency counts of words, otherwise known as **occurrences**, whereas **forms** are a single graphical unit corresponding to several instances in the text [10]. This corresponds to the type/token distinction in Corpus Linguistics. It is also possible to calculate **Repeated Segments** (RS) which returns sequences of at least two consecutive units, or more, that occur several times in the corpus. These objects, forms, occurrences, RS, can be grouped together to create ad-hoc LR generated by the analyst. The resulting resources make up the

analyst's "equipment" providing access to textual tendencies that could otherwise remain hidden by the quantity of data.

2.3 Textometric methods

The Textometric method segments a corpus into comparable zones of text. The news corpora used in the following examples are broken down into smaller groups of articles according to date and in one case according to the writing process (whole article attributed to one author versus user's comments on the article attributed to several authors). Using statistic and probabilistic calculations on the units within each zone, quantifiable information is derived providing the analyst with new knowledge of the textual data. Trends or patterns in the quantifiable information can therefore be observed across the predefined zones.

In this paper, the hypergeometric model[1] is applied to both text zones as well as their forms. In the first case, the calculation shows the statistical probability of a form to appear in a specific zone of the corpus in order to represent the form as having a degree of *specificness* or statistical significance for the zone it appears in. The result is a graphical representation of the *specificness* distribution for the selected forms as will be seen in the figure 1. In the second case, the same calculation is applied directly to a single form, otherwise known as pivot-form, in order to obtain graph or network of interrelated forms from the corpus as a whole or a single zone, table 3. The resulting relations are known as co-occurrences, or the statistical attraction of two or more words in a given span of text (sentence, paragraph, entire article) [14].

Both calculations (*specificness* and co-occurrences) will be used to observe named entities and their various designations over a period of time. Chronological analyses using these methods have already been carried on numerous news media sources ranging from portals such as Lexis-Nexis and Factiva [3] to the French national newspaper Le Monde [15].

In comparison with approaches that use qualitative coding, textual statistics would have a relatively low maintenance cost, due to the minimum amount of actual processing or human development of annotation models. Using relatively simple tokenizers these tools can be applied to a wide range of languages [4].

3 Intelligence Applications

The two use-cases presented here show how the results of Textometric calculations can be used for interpreting vital information from textual data. Whether an analyst is attempting to compile LR for future use or trying to discover relationships that certain points of interest entertain in the data, Textometric methods help shed new light on the intrinsic properties of text elements.

3.1 Named Entity reference detection for opinion mining

The raw material under study comes from a corpus archive of various French online newspapers originally made for commercial purposes of seeking insight on the image of the Socialist Party[2] from November 2008 to August 2009. The data from Le Monde were stripped from the archive for use in the following experiments. The textual data has been automatically extracted from the XML feed for each article and its available user's comments. The main entry of analysis for this kind of task consists in looking for information on NE, particularly political personalities. Typically, here, paraphrases of a NE are valuable because they allow the discovery of semantic variations related to how the NE is perceived. Indeed, whether the focus is on journalistic paraphrases or on nicknames given by Internet users, paraphrases are a major entry point to opinion mining. In the current use-case, this detection step provides candidates for the analysis of how the French President is depicted in web news and is seen through user's conversations.

A first entry to the Textometric analysis consists of using full-text search in the generated dictionary of lexical frequencies, coupled with search based on *regexp*. As a result, one can obtain a list from which derived forms can be selected and grouped in a set. These derived forms are highly informative concerning how a NE is represented in the textual material, with no regards to the linguistic performance of the comment and article writers. A second and complementary entry consists of calculating the *repeted segments (RS)*. The **RS calculations** take text material as input and return an ordered set of objects that can be analyzed contextually. For example, Sarkozy has a number of lexical derivations (*Sarkozyste*, *Sarkoland*) as well as paraphrases (*M. Sarkozy*, *Président de la République*) that help determine the various directions an image analysis must follow. In this case, the lexical derivations for *Sarkozy* portray chiefly a negative image of the President; whereas, the RS show a relatively neutral image that requires further investigation. RS calculations can therefore be used to fruitfully detect NE and how they materialize in the text.

Table 1 – Example of forms and RS extracted sets

Form	*Freq.*	*Repeted Segment*	*Freq.*
Sarkozy	278	Nicolas Sarkozy	85
Sarko	98	de Sarkozy	29
Sarko	19	président de la République	27
Sarkosy	12	de Nicolas Sarkozy	23
Sarkozy	8	M. Sarkozy	19
Sarkozyste	3	de Sarko	16
Sarkozystes	3	Mr Sarkozy	14
Sarkosysme	2	Président de la République	10
Sarkoland	1	le président de la République	9
Sarkoland	1	Le Président de la République	2

[1] The hypergeometric distribution as described in P. Lafon (1980), "Analyse Lexicométrique et recherche des cooccurrences", Cahiers de Lexicologie n°36

[2] The *Parti Socialiste (Socialist Party)* is traditionally on the left-hand political side of the French landscape. The focus is set from November 2008, when a new head of the party is elected, to august 2009, following a deep intern crisis after the defeat at the European Elections.

The above constructed paradigms (lexical derivations, RS) can be seen as a subset creating a new LR. These linguistic phenomena allow the identification of discursive figures that can be contextually analyzed, as shown in [3][15].

Textometric tools allow the analyst to quickly build lexical paradigms. This step is an advantage in itself from a very pragmatic point of view, especially when one is in the position of having to acquire knowledge and accurate linguistic information for building LR from scratch. Transposed to the industrial context, such a process bypasses industrial impediments such as cost-cutting and production time.

The defined paradigms are then set as *Textometric objects*, on which *specificness* calculation can be applied. As seen in 2.3, this **statistical method** is aimed at extracting, for a given **subset of a corpus**, the objects that are over or under represented compared to all the other subsets of the corpus.

Table 2 - Example of extracted paraphrases

Mr Sarkozy	15
Président de la République	10
président de la République	27
Nicolas Sarkozy	85

This kind of results provides the analyst with accurate information on when and **how the *discursive figures* attached to a personality evolve in media news through time**. The shifts of the opinion can be sensed through this evolution, indicating where the attention should be focused, thus acting as a "metal detector" indicating where to dig.

For example, in the articles[3] that make up the corpus, the results (Fig.1) clearly show that the civil paraphrase *Nicolas Sarkozy* and the status paraphrase *président de la République* are highly specific of the Le Monde discourse during intense times of the political agenda (M_904_A to M_907_A), here the European Elections in June[4]. The newspaper discourse itself evolves from April to August, focusing on the person *Nicolas Sarkozy* in April (M_904_A) to emphasizing his status by the segment *president de la République* in June (M_906_A).

On the other hand, in the user's comments of these same articles, the *Président de la République* paraphrase is distinctively over represented in June (M_906_C). The capital letter in the word *Président* is highly informative as indicating a particular attachment to the normative form for writing status words. This paraphrase also evolves into *Mr Sarkozy* in August (M_908_C), while both disappear between June and August. In such cases, it is necessary to go back to the textual material to deliver a more accurate analysis, as specific knowledge of the media situation must be known to interpret the meaning of this trend. Textometric frameworks allow the user to navigate back and forth between statistical results, graphic representations and raw analysis material- the text. It is thus interesting to see the online audience of Le Monde modifying the linguistic material used to refer to *Nicolas Sarkozy*, preferring the latter civil paraphrase *Mr Sarkozy* to the status *Président de la République*. In fact, given that Le Monde is traditionally on the political left side, this shift can be explained by two factors: (i) the rise of provocative or off-topic messages,[5] supporting UMP, Sarkozy's party, resulting

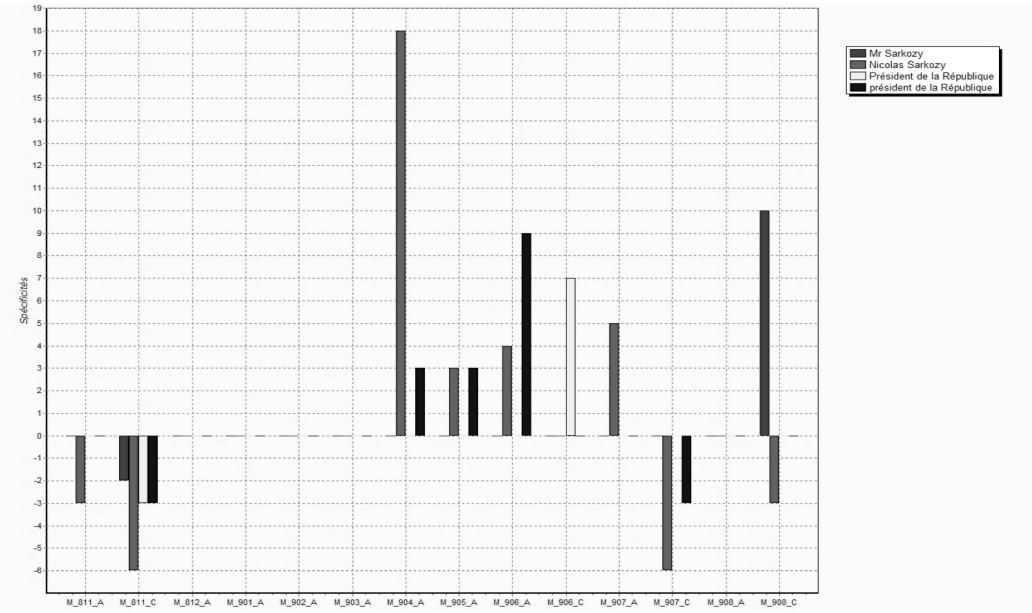


Figure 1 - Monthly variation of *specificness* on the paraphrases for the NE *"Nicolas Sarkozy"*.

[3] It must be specified that the user's comments for some months could not be retrieved due to the fact that Le Monde became a paying newspaper the year we collected the corpus, and though did not provide access to the comments associated to the collected articles.

[4] This European election was punctuated by the defeat of the Socialist Party resulting in violent media confrontations within the party. Nicolas Sarkozy is already President of France at this time.

[5] In Internet slang, this kind of attitude among users is know as a "troll", defining a user who posts provocative or off-topic messages, specifically in discussion forums.

in a *specificness* peak for *Président de la République*; (ii) the rise of unsatisfied Socialist Party supporters stemming from media confrontations in the party after their defeat. Through these confrontations, Sarkozy's position is reinforced in the political arena, resulting in a *specificness* peak for *Mr Sarkozy*. This segment is far from completely neutral for image interpretations as it seems to remove Sarkozy from his status as President without going so far as to use insulting paraphrases found in other French newspapers at the time.

3.2 Named Entities and current events for business intelligence applications

As demonstrated in 3.1, Textometry derives new information from the trends in the various forms of an NE, but how can we access information that the analyst is not specifically looking for? This is akin to the problem discussed above, standard information extraction techniques use qualitative coding to derive interpretations of the data [6]. This leads to potentially missing unknown information, in other words, semantic annotations provide only as much enriched content as the resource has been designed for.

This second use-case follows the Firthean inspiration [7]: "You shall know a word by the company it keeps", where a great deal of research has been done on lexical affinities (collocations or co-occurrences) between words. Here co-occurrences is understood as the statistical attraction of two or more words as discussed in 2.3. This calculation allows for the precise description of the lexical environment of a pivot-form through several variables left up to the analyst: (i) **co-frequency**, indicating the lowest number of times two words must appear together in the corpus to be considered as a co-occurrence; (ii) **threshold**, designating the probability level that a co-occurrence relationship must have to be considered; (iii) **segmentation context**, giving the punctuation boundary for the pivot-form, sentence, paragraph, other. What results is a list or network of co-occurring forms that can be interpreted depending on their statistical attraction (table 3).

The hypothesis here is that as a current event is discussed in the media, the lexical network produced by the co-occurrence calculation will be greater during an event than during periods of calm or low activity of the NE. This is similar to a sort of "buzz" effect where it has been shown that the more an NE is discussed by the media, the more likely it is that an event involving the NE is taking place [12]. However, the frequency of an NE alone may not be enough information to determine if an event may be taking place at a given time in the data. The high frequency of an NE could simply denote a popular topic. Two factors are thus important to discovering events, (i) the lexical network and (ii) chronological trend in the data.

The corpus used for this study is a sub-section of articles from the NYT Annotated corpus [19]. The articles correspond to Business/Financial Desk from 2001-2002 and were stripped of the xml to be put into txt format for more efficient analysis by Textometric tools. In a method similar to [12] and [4] the co-occurrence network for an NE can be calculated month by month, showing emerging information through the resulting network. In comparing both the variations in the number of different co-occurrences produced using the same input criteria, as well as the lexical units themselves, it is possible to determine what events an NE is involved in at a given time. The following example shows the monthly trend in articles mentioning the NE *Xerox* from January 2001 to December 2002, which corresponds to 160 articles in the NYT.

As can be observed in the distributions in figure 2, the form *Xerox* fluctuates greatly over the course of two years. These peaks in the number of occurrences show potential zones of interest for this NE for the periods of Febuary-March 2001 and April-July 2002. This "buzz" corresponds to the accounting scandal *Xerox* was involved in with the firm KPMG. When studying the distribution of number of different co-occurrences (each interrelated form counts as a single co-occurrence), a sharp peak can be seen for the month of April (33 co-occurrences), meaning the lexical network is much more abundant for this period and that, in following the hypothesis, an event may be taking place.

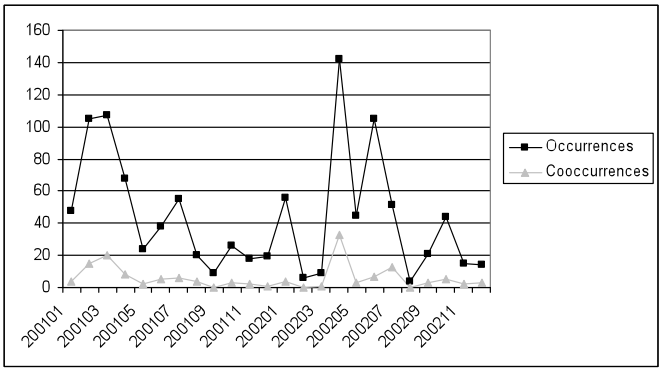


Figure 2 – Monthly variation of the number of occurrences for the NE and the number of co-occurrences for the pivot-form *Xerox*

To verify this idea, the lexical network for the month of April was compared to other months. April shows a higher number of *unexpected* vocabulary relating to the NE *Xerox* (table 3), in other words few co-occurrences actually describe the NE (*leases*, for example).

The majority of co-occurrences for this month are in keeping with the complaint filed against *Xerox* by the SEC in April 2002 (*complaint*, *kpmg*, *revenues*, *1997*, *accounting*).

Table 3 – Co-occurrences for Xerox, April 2002, co-freq 5, threshold 5

Form	Frequency	Co-Freq	Specif	Context
kpmg	100	25	23.43	19
complaint	18	9	12.68	9
pay	184	12	5.47	11
leases	23	5	5.63	5
numbers	55	7	5.66	7
that	2606	71	6.29	49
corporation	116	11	6.68	11
future	56	7	5.61	4
fine	42	10	10.15	9
its	882	49	14.44	38
restate	24	10	12.92	9
securities	134	12	6.88	12

revenue	98	10	6.51	10
cents	87	8	5.20	8
1997	69	9	6.92	9
revenues	27	5	5.28	5
accounting	329	32	16.52	30
earnings	138	12	6.74	10
share	116	13	8.46	11
agreed	39	9	9.17	9
method	20	5	5.95	4
investigation	91	12	8.74	10
commission	126	16	10.85	16
had	635	27	6.47	24
it	1406	51	8.43	37
settlement	29	7	7.62	7
auditor	25	5	5.45	5
financial	276	17	6.72	17
filed	61	9	7.38	9
settle	16	5	6.48	5
exchange	93	13	9.65	13
results	77	15	13.03	14

When analyzing the depleted lexical networks for the other months (on average between April 2001 and March 2002, only 3 co-occurrences are found using the same criteria), there is much more *expected* vocabulary: *computing, sales, services, representatives*, for example. In a manner similar to the use-case discussed in 3.1, co-occurrences are used here as the 'metal detector' for finding potential events that involve the NE. However, it remains necessary for the analyst to determine what vocabulary can be *expected* for a given NE.

4 Discussion

In this paper we compared two intelligence application use-cases applying Textometry as a method for mining information on named entities present in online news articles from Le Monde and the New York Times. Named Entities were used as an entry point for the analyses of the corpora and Textometric tools provided results for creating new Linguistic Resources as well as identifying relationships with other entities. These methods use quantitative information to formulate qualitative interpretations and thus can be included among other text mining strategies.

Both use-cases illustrate how Textometry can help media analysis tasks through two different, but complementary approaches (*specificness* and co-occurrence analysis). In a more industrial context, the analyses presented here yield promising results for business and communication intelligence applications. Three main contributions are established here:

- corpus-driven Linguistic Resource building and adaptation or update by using the Repeated Segments and lexical derivation exploratory functions of Textometric tools ;
- identification of trends with *specificness* calculation to detect over or under represented segments in a subset of the corpus in order to guide qualitative analyses of current events ;
- chronologically emerging information through the co-occurrence network of a specific NE to target zones of activity or events.

These points can help analysts in the desicion making process by shedding light on evolving trends in the corpus and potential critical information.

However, this method should be distinguished from other robust NLP approaches due to the important emphasis on the role of the user. Contrary to other mining techniques, this approach is not fully automated which raises interoperability issues with other computer processing tasks. Textometry demands the "return of the expert in the system". This explains why it is often not inculded among commercialized applications.

For future research, several venues must be explored:

- evaluating the interoperability of Textometric tools with other robust NLP applications. A combined approach using both Textometry and precoded information requires further experimentation ;
- confronting results acquired with Textometric methods against results obtained through NLP methods such as building ontologies for opinions or NE-relationship extraction ;
- analyzing the results obtained with different NLP applications such as tokenizer or syntactic taggers through Textometric methods.

In sum, deriving knowledge from corpora without predefined information models, often provided through qualitative coding, is easier said than done. This paper demonstrated how such annotations can be skirted with statistical calculations and Textometric methods, cutting production time. These methods provide adequate functions enabling interaction between the expertise of the user and the processing tools. The analyst, therefore, can achieve more in depth research.

References

[1] Bloom K., Stein S. & Argamon S., Appraisal extraction for news opinion analysis at NTCIR-6, Proceedings of NTCIR-6, 2007, p 279-289.

[2] Bollier, D. The Promise and Peril of Big Data. Washington, DC : The Aspen Institute, 2010.

[3] Delanoë, A. 2010. Statistique textuelle et series chronologiques sur un corpus de presse écrite. Le cas de la mise en application du principe de précaution. Proceedings, JADT'2010.

[4] Delaplace R., Leenhardt M. & Wu L-C., Méthode de conception d'une application de veille et d'Analyse Linguistique Assistée par Ordinateur, VSST Conference, Toulouse, France, 2010.

[5] Fayyard, U.M, Piatesky, G., Smyth, P. & Uthurusamy, R. Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press, 1996.

[6] Feldman R. & Sanger J., The Text Mining Handbook : Advanced Approaches in Analyzing Unstructured Data, Cambrigde University Press, 2006, 422 p.

[7] Firth, J.R. A Synopsis of Linguistic Theory 1930-1955, Linguistic Analysis Philological Society, Oxford, 1957.

[8] Grishman, R. & Sundheim, B. Message Understanding Conference- 6 : A Brief History. Proceedings of the 16th International Conference on Computational Linguistics (COLING), I. Kopenhagen, 1996 p.466–471,.

[9] Kodratoff, Y. Knowledge discovery in texts: A definition and applications, Proceedings of the International Symposium on Methodologies for Intelligent Systems, 1999, volume LNAI 1609, p. 16–29.

[10] Lebart, L. & Salem, A. Statistique textuelle. Paris, Dunod, 1994.

[11] Lent, B., Agrawal, R., & Srikant, R. Discovering trends in text databases, Proceedings KDD'1997, AAAI Press, 14–17 p. 227–230.

[12] MacMurray E. & Shen L., Textual Statistics and Information Discovery: Using Co-occurrences to Detect Events, VSST Conference, Toulouse, France, 2010.

[13] Martin J.R. & White P.R.R., The language of evaluation: appraisal in English, Palgrave, London, 2005.

[14] Martinez, W. Contribution à une méthodologie de l'analyse des cooccurrences lexicales multiples dans les corpus textuels, Thèse pour le doctorat en Sciences du Langage, Université de la Sorbonne nouvelle - Paris 3, 2003.

[15] Née, E. Insécurité et élections presidentielles dans le journal Le Monde, Lexicometrica numéro thématique « Explorations Textuelles », S. Fleury, A. Salem. 2008

[16] Poibeau T. Extraction automatique d'information. Du texte brut au web sémantique. Paris : Hermès Sciences, 2003.

[17] Poibeau, T. Sur le statut référentiel des entités nommées, Proceedings TALN'05. Dourdan, France, 2005.

[18] Salem A., Introduction à la résonance textuelle, In Actes des JADT 2004 (7 èmes Journées internationales d'Analyse Statistique des Données Textuelles), 2004, p 986-992.

[19] Sandhaus, E., The New York Times Annotated Corpus. Philadelphia: Linguistic Data Consortium, 2008.

[20] Tufféry, S., Data mining et statistique décisionnelle: l'intelligence des données. Paris : Editions Technip, 2007.

[21] Stoyanov, V., Cardie, C., Litman, D. and Wiebe, J. Evaluating an Opinion Annotation Scheme Using a New Multi-Perspective Question and Answer Corpus. Working Notes of the 2004 AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications, 2004

[22] Wilson, T., Ruppenhofer, J., Wiebe, J., Documentation for MPQA Corpus version 2.0, [online] http://www.cs.pitt.edu/mpqa/databaserelease/Database.2.0.README Date consulted: May, 12th, 2011

[23] Wright, K., Using Open Source Common Sense Reasoning Tools in Text Mining Research, the International Journal of Applied Management and Technology, 2006 vol 4 n°2 p.349-387.

Hypernodes in the UNL Interlingua

I. Boguslavsky
Departamento de Inteligencia Artificial, Universidad Politécnica de Madrid, Madrid, Spain

Abstract - *Universal Networking Language (UNL) was launched by the Institute of Advanced Studies of the United Nations University for the facilitation of multilanguage and multiethnic access to communication networks such as WWW. UNL is used to support different types of applications, from multilingual generation to abstracting and cross-lingual information retrieval. The main representation unit of UNL is a graph formed of UNL lexical nodes, semantic relations and attributes. One of the important features of UNL is the fact that a graph may contain hypernodes, i.e. connected subgraphs that function as single nodes. We describe properties of hypernodes and present some linguistic phenomena for which they can be used..*

Keywords: Universal Networking Language, UNL, hypernode, scope, interlingua, sentential adverbial.

1 Introductory remarks

The Universal Networking Language (UNL) developed by H. Uchida at the Institute for Advanced Studies of the United Nations University is a meaning representation language designed for multilingual communication in electronic networks, information retrieval, summarization and other applications.

The philosophy underlying UNL language design and the current state of system development are discussed in the publications by Hiroshi Uchida, the author of UNL, which, along with other relevant data, can be found at the UNL official site *http://www.undl.org*. Recent developments in UNL are outlined in [1]-[4]. In Section 2 we will give a brief outline of UNL, and in Section 3 we will dwell on one of its key elements, the hypernode.

2 What is a UNL expression?

UNL is a computer language intended to represent information in a way that allows generating a text expressing this information in a very large number of natural languages. Formally, an expression in UNL is an oriented hypergraph that corresponds to a natural language sentence in the amount of information conveyed. The arcs of the graph are interpreted as semantic relations, such as agent, object, time, reason, etc. The nodes of the graph may be simple or compound. Simple nodes are special units, the so-called Universal Words (UWs). A UW roughly corresponds to a concept. It consists of a Headword, which is usually an English word or phrase, and a set of constraints that modify the meaning of the Headword or supply it with additional information. A compound node (hypernode) consists of several simple or compound nodes connected by semantic relations. In addition to propositional content ("who did what to whom"), UNL expressions are intended to capture pragmatic information such as focus, reference, speaker's attitudes and intentions, speech acts, and other types of information. This information is conveyed by means of attributes attached to the nodes, e.g. *@imperative, @generic, @future, @obligation.*

To give a general idea of what a UNL expression looks like, we will use an example of average complexity and comments on some of the UNL features it demonstrates[1].

(1) *The actress whom we wanted to see so much did not appear on the stage in the first act.*

UNL expression of (1):

```
[S:1]
{org:el}
The actress whom we wanted to see so
much did not appear on the stage in the
first act.
{/org}
{unl}
mod(act(icl>dramatic_composition).@def,
first(icl>adj, ant>last))
agt(appear(icl>come>do,agt>thing,plc>
thing).@not.@past.@entry, :01)
plc(appear(icl>come>do,agt>thing,plc>
thing).@not.@past.@entry,
stage(icl>platform>thing).@def)
```

[1] Except for this expression and the last one in 3.5, all other examples of UNL expressions will be simplified.

```
tim(appear(icl>come>do,agt>thing,plc>
thing).@not.@past.@entry,
act(icl>dramatic_composition).@def)
obj:01(see(icl>perceive>be, aoj>thing,
obj>thing, cob>uw):50,
actress(icl>actor>thing):39.@def.@entry)
man:01(want(icl>desire>be,cob>uw,obj>uw,
aoj>thing):Q0.@past,
so_much(icl>how):OR)
aoj:01(want(icl>desire>be,cob>uw,obj>
uw,aoj>thing):Q0.@past,we(icl>person))
obj:01(want(icl>desire>be,cob>uw,obj>
uw,aoj>thing):Q0.@past,see(icl>perceive>
be, aoj>thing, obj>thing, cob>uw):50)
{/unl}
[/S]
```

A simplified but convenient form of visualizing this expression is shown in Fig. 1. For brevity, UWs are contracted in this picture.

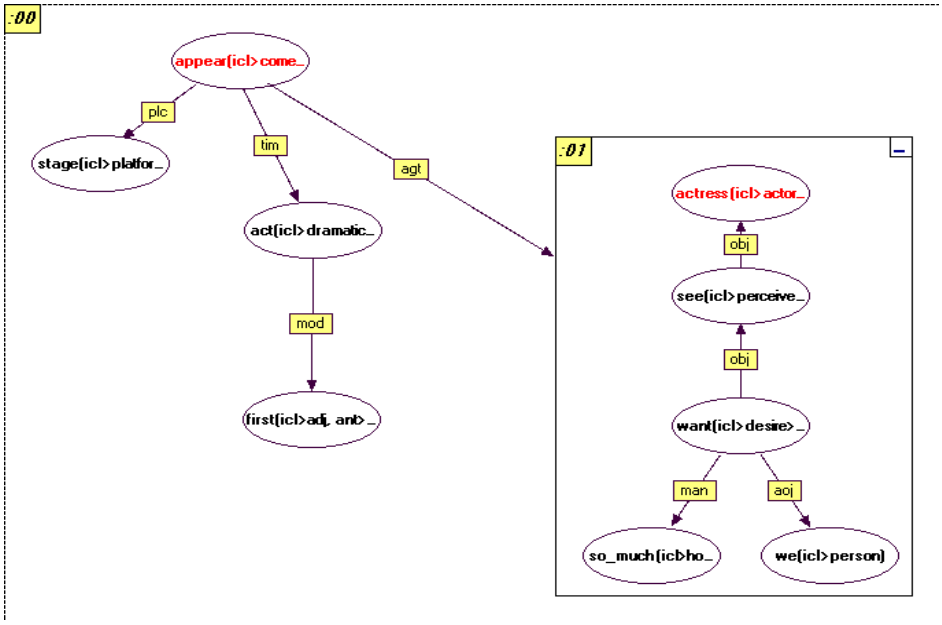


Fig. 1. Graphic representation of the UNL expression of (1)

A UNL expression of a sentence is a set of relations. Each relation connects two UWs. In the expression above, the following relations are used: `agt` (agent), `obj` (object), `aoj` (subject of a state), `man` (verb modifier), `mod` (noun modifier), `tim` (time), and `plc` (place). Most of the UWs are supplied with constraints (expressions in brackets following the headword). Constraints mostly serve to indicate the semantic class of the UW and to restrict the meaning of the headword. For example, the constraint `icl>come>do` of *appear* identifies the relevant meaning of *appear* and at the same time shows that it is an action, as opposed to a process or a state. *First* has two constraints: `icl>adj` says that *first* is an adjectival concept and `ant>last` shows that it is the antonym of *last.* The latter constraint helps to differentiate this meaning of *first* from another one (cf. *first violin*) where the antonym is not *last,* but *second.* Attribute `@entry` marks the "main" element of the structure (called "entry node") which normally corresponds to the syntactic top node of the sentence or a corresponding part of the sentence. Attribute `@def` ascribed to *stage* shows that it has a definite reference.

The UNL expression of (1) contains a hypernode that corresponds to the phrase *the actress whom we wanted to see so much,* which consists of a noun modified by a relative clause. The hypernode has a numeral identifier (`:01`) which is used to represent the hypernode in relations (as in `agt(appear,:01)`) and to mark relations it contains (as in `aoj:01(want, we)`). In the next section, we will present our current view of hypernodes and list some linguistic phenomena which are represented in UNL by means of hypernodes.

3 Hypernodes

3.1 Types of hypernodes

The basic idea behind the introduction of a concept of hypernode is that it is a set of nodes that functions in a certain sense as a single node. This means that a hypernode as a whole can do whatever a simple node does, namely, it may participate in relations, both incoming and outgoing, and have its own attributes.

There are two major types of hypernodes in accordance with the task they are solving: semantically loaded and syntactic ones.

Semantically loaded (to be henceforth called semantic, for short) *hypernodes* are those that convey information which differentiates meanings. If such a hypernode is not produced, some information is lost, meaning distinction is not made and the result of the deconversion may be different. Therefore, semantic hypernodes are obligatory. Example:

(2) *culture and communication in the Internet.*

(2) has two meanings: (a) both culture and communication are in the Internet or (b) only communication is in the Internet. The first of these meanings requires that `culture` and `communication` be enclosed in a hypernode, to which the `Internet` node should be connected.

```
(2a) and:01(communication, culture)
     plc(:01, Internet)
(2b) plc(communication, Internet)
     and(communication, culture)
```

If we fail to make a hypernode, we lose an important piece of information.

Semantically unloaded (or, syntactic) *hypernodes* are those that basically convey information on syntactic composition of the sentence. They do not differentiate between meanings but serve for convenience and easier comprehension of UNL graphs. For human UNL readers, it is easier to understand a complex UNL graph if it is segmented by hypernodes into meaningful components. They may correspond to clauses, large noun phrases, large coordinated strings, etc. If such a hypernode is not produced, the structure of the UNL graph will be less transparent and more difficult for the humans to grasp but no loss of information occurs and the deconversion result is the same. Therefore, syntactic hypernodes are optional.

Syntactic hypernodes can be exemplified with sentence (3).

(3) *The state should take appropriate steps to return any cultural property imported after the entry into force of this convention, provided, however, that the requesting state shall pay just compensation to an innocent purchaser who has valid title to that property.*

In this sentence, at least three syntactic hypernodes can be established, which enclose:

(a) the complex NP consisting of NP *any cultural property* and participial clause *imported after the entry into force of this convention,*
(b) the adverbial clause *provided, however, that the requesting state shall pay just compensation to an innocent purchaser who has valid title to that property,* and
(c) the complex NP within (b) consisting of NP *an innocent purchaser* and relative clause *who has valid title to that property.*

3.2 Impermeability of hypernodes

It is natural to require that elements of the hypernode, as opposed to the hypernode itself, should not have links with UWs outside the hypernode. Otherwise, the claim "the hypernode functions as a single node" will lose its sense. However, strictly speaking, this is not always true. A hypernode may function as a single node in one respect but not in another.

Cases (2) and (3) discussed above are clear examples of impermeable hypernodes. Links from the outside of the hypernode do not penetrate inside, and the other way round. The impermeability condition is violated when the hypernode is used to convey meta information, such as emphasis or formatting. The only way to say that UWs A and B should be put in parenthesis or quotes or printed in italics is to enclose A and B in a hypernode and assign a corresponding attribute to it. For example, sentence (4) contains a title marked by double quotes. In (4a) a hypenode `:01` is made which comprises the subgraph corresponding to the phrase *towards linguistic and cultural diversity in cyberspace* and is assigned the `@double_quote` attribute:

(4) The paper is entitled *"Towards linguistic and cultural diversity in cyberspace"*.

```
(4a) obj:01(towards, diversity)
     mod:01(diversity, cultural)
     and:01(cultural, linguistic)
     plc:01(diversity, cyberspace)
     :01.@double_quotes
```

Hypernodes set up to convey this type of information can be permeable, since semantic links can easily cross the borders of phrases marked by double quotes, brackets, intonation mark-up and similar devices. For example:

(5) *John "decided to educate" his parents.*

Here, there is a semantic link between *educate* and *parents*. Although this is a link going from the inside of the hypernode to the outside, such a hypernode still conforms to the basic idea of the hypernode, since the phrase *decided to educate* does behave as a single node – with respect to the double quotes.

3.3 How to observe hypernode impermeability.

There is a large class of situations when an object participates in more that one predication within the sentence. Normally, it means that there are several

relations in which it participates. In (6) John is the agent of two actions – *decide* and *go away*.

(6) *John decided to go away.*
(6a)
```
agt(decide, John)
obj(decide, go away)
agt(go away, John)
```

However, if one of these predications makes a hypernode, there appears a cross-hypernode link (shown in bold):

(7) *John demanded that Bill apologize to him.*
(7a)
```
agt(demand, John)
obj(demand, :01)
agt:01(apologize, Bill)
obj:01(apologize, John)
```

A way to avoid violation of the impermeability requirement consists in postulating different but co-referential nodes:
(7b)
```
agt(demand, John:A1.@ref1)
obj(demand, :01)
agt:01(apologize, Bill)
obj:01(apologize, John:A2.@ref1)
```

3.4 Linguistic phenomena described with hypernodes.

Below we will briefly list several linguistic phenomena that are described by means of hypernodes. For lack of space, we will not give UNL graphs, but will merely put the corresponding fragment of the sentence in brackets and provide minimal comments.

1º Clause marking.

Cf. example (3) above.

2º Adjective placement.

so-called (moral preoccupations)

A hypernode is needed to show that *so-called* is related to the whole phrase *moral preoccupations* and therefore cannot split it.

3º Common dependents.

Cf. example (2) above.

4º Attribute assigned to a fragment of the sentence.

Food is cheap in England,(I believe)

A hypernode is used to show that a phrase (*I believe*) is used parenthetically. The hypernode is assigned the `@parenth` attribute.

5º Scope of negation.

(They didn't sleep until midnight)

The sentence permits two interpretations:

a) 'they started sleeping at midnight'; no hypernode.

b) 'not that they slept until midnight, they woke up earlier'; a hypernode encloses the whole sentence and is assigned the negation attribute.

6º Sentence adverbials.

He answered the dean very cleverly (his answer was very clever; no hypernode).

Very cleverly, (he answered the dean) (it was very clever of him to answer the dean; hypernode needed).

3.5 Some complex examples of hypernodes.

We will give two examples which have several hypernodes of different types.

(8) *This fall has not been so rainy and wet as last fall was.*

This sentence is a negation of the sentence

(8a) *This fall has been as rainy and wet as last fall was.*

Therefore, sentence (8a) should be made a hypernode, which constitutes the scope of negation. Within this hypernode, there is another one, composed of the coordinated string *rainy and wet*. The reason for introducing this hypernode is that both predicates have common dependents.

```
aoj:01(:02, fall:01)
and:02(wet, rainy)
man:01(:02, as_as)
bas:01(as_as, fall:02)
aoj:01(:02, fall:02)
:01.@not
```

Note that *so...as* in sentence (8) is a variant of *as...as*, which appears under negation. Since we extracted negation from hypernode `:01`, we have to replace *so...as* with *as...as.*

Note also that sentence (8) contains 2 occurrences of *fall,* which are not co-referential. This is the reason why they are supplied with different numerical IDs in the UNL extression.

(9) *I wonder whether you should go and see her or whether it is better to write to her.*

This sentence is represented by means of four hypernodes. The first hypernode (:01) is a convenience one: it serves for marking a complex embedded clause: *(whether) you should go and see her or (whether) it is better to write to her.*

Hypernodes :02 and :03 embrace phrases connected by *or*. These hypernodes bear their own attribute each - `@interrogative`. Finally, hypernode :04 includes the phrase *should go and see her.* The hypernode is needed to show that nodes `go` and `see` have a common dependent – `you`.

The UNL expression of this sentence is given below and its visualization is presented in Fig. 2.

```
[S:1]
{org}
I wonder whether you should go and see
   her or whether it is better to write
   to her.
{/org}
{unl}
obj(wonder(icl>inquire>be,obj>uw,aoj>
   person).@present.@entry,:01)
aoj(wonder(icl>inquire>be,obj>uw,aoj>
   person).@present.@entry,i(icl>person
   ))
or:01(:02.@entry.@interrogative,:03.
   @interrogative)
aoj:02(good(icl>adj,ant>bad).@present.
   @entry,write(icl>communicate>do,cob>
   uw,agt>thing,obj>thing,rec>person))
man:02(good(icl>adj,ant>bad).@present.
   @entry,more(icl>how))
gol:02(write(icl>communicate>do,cob>
   uw,agt>thing,obj>thing,rec>person),
   she (icl>person):02)
agt:03(:04,you(icl>person).@pl)
and:04(see(icl>perceive>be,cob>uw,obj>
   thing,aoj>thing).@entry.@need,
   go(icl>move>do,plt>place,plf>place,
   agt>thing).@present)
obj:04(see(icl>perceive>be,cob>uw,obj>
   thing,aoj>thing).@entry.@need,she
   (icl>person):01)
{/unl}
[/S]
```

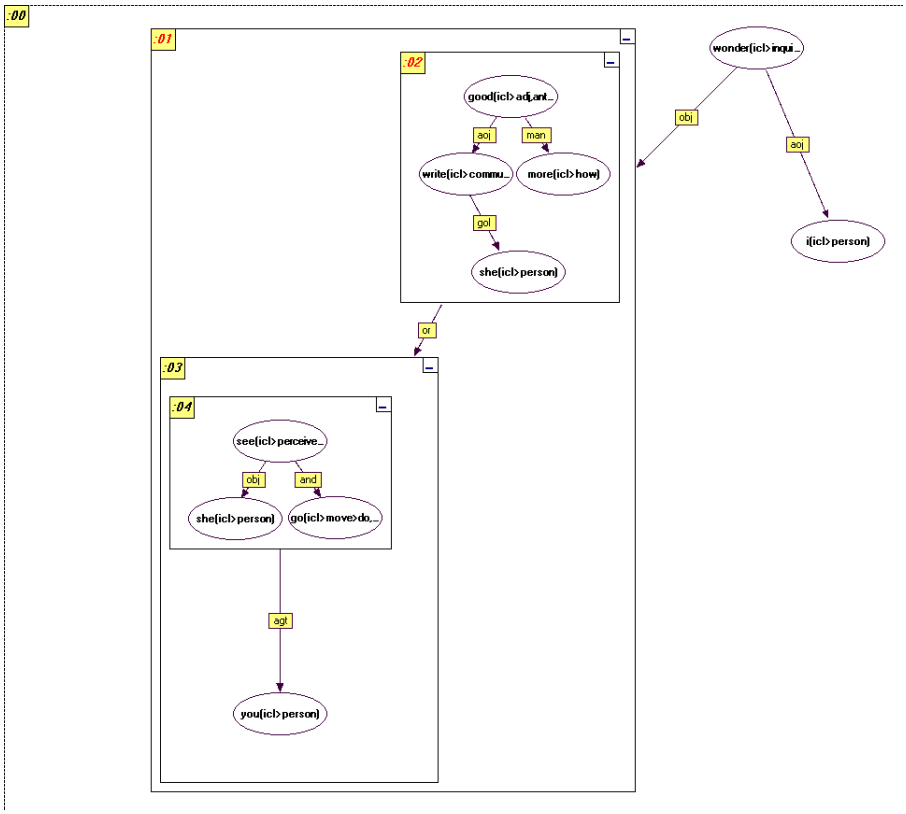


Fig. 2. Graphic representation of the UNL expression of (9)

4 Conclusions

In natural languages, there is a wide range of situations in which some phrases and word combinations behave as a single word. UNL has a formal means to represent this type of behavior – hypernodes. These are connected fragments of a graph that function in certain respects as single nodes. Our purpose in this paper was to discuss the properties of hypernodes and present some linguistic data, which can be adequately described with hypernodes[2].

[2] The work reported in this paper has received partial funding from the Russian Foundation for Humanities (grant No. 10-04-00040a) and the Russian Foundation for Basic Research (grant No. 11-06-00405), which is gratefully acknowledged. The author is grateful to Etienne Blanc and Vyacheslav Dikonov. Discussions with them were very useful.

5 References

[1]Boitet, C. A Rationale for using UNL as an Interlingua and more in various domains. Proceedings of the First International Workshop on UNL, other Interlinguas and their Applications. International Conference on Language Resources and Evaluation (2002).

[2] Boitet C., Boguslavskij I. & Cardeñosa I. An Evaluation of UNL Usability for High Quality Multilingualization and Projections for a Future UNL++ Language. In "Computational Linguistics and Intelligent Text Processing (Proc. CICLING-2007)", A. Gelbukh, ed., Springer (LNCS 4394), 2007, pp. 361-373.

[3] Boguslavsky I. Some controversial issues of UNL: linguistic aspects. Universal Networking Language: Advances in Theory and Applications. Jesús Cardeñosa, Alexander Gelbukh, Edmundo Tovar (Eds.) Special issue of Research on Computing Science, IPN, 2005, p.77-101.

[4] Boguslavsky I., J. Cardeñosa, C. Gallardo, L. Iraola, The UNL Initiative: An Overview. Lecture Notes in Computer Science, Volume 3406, 2005, p. 377 – 387, Springer Berlin / Heidelberg.

Lexical and Semantic methods in design of the Problem-oriented Linguistic resources

Olga Kozhunova[1]
[1] Institute for Informatics Problems, The Russian Academy of Sciences,
Moscow, Russia

Abstract **-** In the paper up-to-date methods and approaches to lexical and semantic modelling are analyzed, namely, in the works of such prominent specialists in the field as N. N. Leontjeva, E.V. Paducheva, J.D. Apresyan, R. Jackendoff, J. Sowa, and others. Some of the linguistic resources (integrated into information systems) with semantic- oriented features and other language-specific attributes are considered. Apart from the analyzing and reviewing of the existing approaches and resources, some suggestions and additions are given by the author, including the approaches of the artificial intelligence techniques and JSM-method formalisms.

Keywords: Lexical and Semantic modelling, Linguistic resources of the information systems, JSM-method.

1 Introduction

Last decades research preferences emphasize the significance of computer linguistics methods and models applied to various interdisciplinary tasks. Modelling of natural language mechanisms for problem-oriented systems touches up the overall quality of information systems and their performance, especially when it comes to the semantic aspect of text processing.

The above mentioned methods and models are used primarily for knowledge representation, formalization of their structures and entities, and adjustment to facilitate its automatic processing. Today the majority of information systems includes semantic component integrated into their linguistic software. Among them are well-known WordNet, EuroWordNet, Ontolingua, Russian dictionary Ruslan, etc.

In the paper lexical and semantic methods are viewed from the perspective of design and development of the problem-oriented linguistic resources of the domain-specific information systems. By the methods we mean a complex of theoretical terms designed to analyze and acquire basic semantic units from natural language texts at lexical and phraseological level. An effort to generalize the existing methods of this kind and to suggest a new enhancing approach is made.

Though the methods mentioned above are characterized as theoretical ones, nowadays they imply plenty of essential applications, namely, building of coordinated systems of domain-specific terms, analysis and formation of links between various lexemes, modelling of semantic relations at lexical and phraseological levels of any language, classification and typology of the domain-specific concepts, private reference tasks, coordination of various concepts understanding among experts of a field or domain, semantic component analysis, etc. This implies plenty of possible research directions and benefits when dealing with semantic and lexical methods integrated into linguistic resources.

2 Lexical and Semantic methods

Lexical and Semantic methods are applied in diverse fields and tasks. For instance, according to N.N. Leontjeva, almost every automatic text interpretation involves need in dictionaries that refer text units as symbol objects to their semantic equivalents [1, 2]. Among the dictionaries capable of such function is automatic semantic dictionary RUSLAN (Russian Dictionary for Analysis) designed and developed at The Moscow State University, Russia [1].

In RUSLAN Semantic information is encoded by specific descriptors indicating semantic category of a word considered. In total, 120 descriptors divided into two subsets are used. The first one is meant for Semantic Attributes (SA) description including such of the type ARTEFACT, SUBSTANCE, SITUATION, INFORMATION, etc. For example (1):

SA("electricity") = PHENOMENON & ENERGY (1)

The second subset contains descriptions of binary semantic relations (SR) of the type PART(A,B), FORM(A,B), BELONG(A,B), etc. Descriptors may construct logic forms – conjuctions and disjunctions. As a rule, conjunctions combine generic and specific in the lexical meaning. They help for modelling of component semantic word structure [2, 3]. The Dictionary functions as a data base, new concepts/relations are entered there in specific patterns predetermined in the system [3].

Apart form the above mentioned, there is an approach to Lexical and Semantic modelling of the natural language processes which has been implemented in the system Lexicograph [4]. Lexicograph is a semantic dictionary also designed as a data base. Traditional dictionaries indicate a list of separate meanings for each word they contain. In order to reestablish the semantic unity of a word, Lexicograph formulates two interdependent tasks [5]:

- Present each given word meaning in the form allowing for getting a clear explanation of the language behavior of the word within its specific meaning;
- Visualize links between words, i.e. build a hierarchy of words meanings, or even a paradigm of meanings applicable to a certain category of words.

The system gives formats for presenting basic meanings types, and its user may edit the base entering new words and meanings. The current version is devoted to the category of words "Russian verbs" [5].

E.V. Paducheva is the principal ideologist of the system [4, 5]. She claims reestablishment of a word unity as one of the urgent tasks of semantics as a discipline. Up-to-date research techniques require a given word to be parsed into separate lexemes and postulate polysemy which speakers seldom perceive. In order to solve the polysemy problem Paducheva and her colleagues are guided by Kurilovich approach [6]. It concerns the concept of derivation and description of derivative meanings with the help of the original one.J.D. Apresyan, the leading expert in the field of lexical semantics and the leader of the Moscow Semantic School [7], states that today semantics is a specific component of the complete language description regarded as a formal tool which is modelling language behavior of humans. He also assumes that Lexical and Semantic modelling is merely modelling of the language skills rather than of knowing reality. These two assumptions are usually taken into account when implementing some applications of Computational linguistics.

Apresyan claimed one of the basic semantic rules regulating correct understanding of a text by a listener as one more postulate of lexical semantics. He formulated it, as follows: one chooses such understanding of a sentence given which is characterized by maximal repetitiveness of the semantic elements. In other words, this postulate (The Rule of Semantic Coordination) states that the meaning of a polysemic word is getting obvious from the context [8,9].

Theoretic research made by Apresyan was applied in the model "Meaning<->Text" (by I.Melchuk and A.Zholkovski) when he designed Explanatory and Combinatorial Dictionary. It became a dictionary of a new type due to its capability of demonstrating nontrivial combinations of lexemes. Он становится словарём нового типа, поскольку отражает, прежде всего, нетривиальную сочетаемость лексем. Words semantics in the dictionary is described as expanded formalized explanations involving only limited set of units. More complex elements (in semantic sense) are explained through common ones until it comes to "semantic primitives". This approach of complex semantic description had much in common with the principal concept of the Polish Semantic School headed by A. Wezbicka and A.Boguslavsky [10]. Apart from the dictionary mentioned Apresyan has designed plenty of other applications of his theories, e.g. New Explanatory Dictionary of Synonyms, Linguistic software of the ETAP system (machine translation), etc. [11, 12].

Among other prominent experts in the field of lexical semantics, cognitive linguistics, and generative linguistics are Ray Jackendoff and John Sowa. Notably, the latter being not linguist but a specialist in the field of Informatics and IT-technologies, made a significant contribution to the domain of Knowledge representation and Knowledge acquisition from natural language structures.

As regards Ray Jackendoff, he constructed the theory of semantic form expression and a comprehensive theory on the foundations of language [13-15].

One of the main goals of the theory suggested is characterization of the conceptual elements (e.g., a word and a sentence), thus constructing an explanatory semantic representation to conduct semantic analysis of the language units [13]. Jackendoff suggested to assume as the basic postulate that meaning in natural language is an information structure encoded by people at the mental level. This postulate enables us both to extend. boundaries of language representations and meaning acquisition in any language, and to apply theory of conceptual semantics in the fields studying problems of conceptualization of higher orders (logic, psychology, behavior science, etc.). In respect to linguistic tasks and formalization of its representations and processes this theory allowed of systemizing of language expressions in terms of semantic, or conceptual structures [13, 14]. According to Jackendoff, there is no need in interpretation of a conceptual structure by a structure of other level, since expressions of the conceptual structure are interpretations themselves. That is, the question is in the choice of the structures formalization and representation tools. For example, if an entity C from the real world has no representation in the mind of a person X (no mappings at the mental level), it involves that C either doesn't exist for X, or C is unavailable for X. It means that without mental representation of C person X wouldn't be able to address it as an expression, a language structure. Therefore one of the disambiguation methods is constructing mental relations with outer structures that already possess features necessary for formalizing of the entity C representation [15].

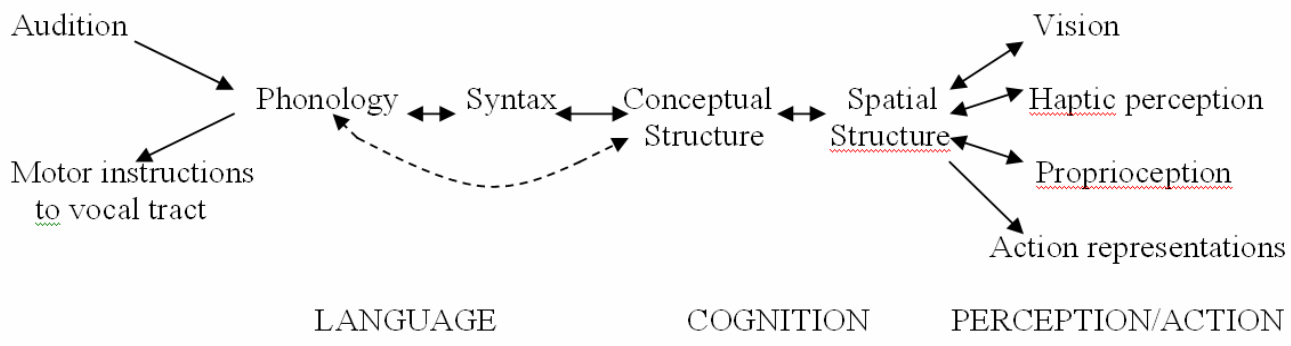


Figure 1. Architecture of the mind (according to R. Jackendoff)

Thus, Jackendoff considers that conceptual semantics, as linguistics in general, can't exist autonomously but closely relates to all the spheres of human perception which allows him/her of outworld comprehension and participation in various types of activities (Figure 1). This idea is very close to what Apresyan's and Wezbicka's ideas mentioned above. Moreover, such approach to Semantic modelling leads us to abstractions of higher level when studying lexical and semantic relations and concepts in natural language. To our mind, it happens by reason that context and semantic primitives are not only words, phrases and even large fragments of text encircling an expression/sentence under analysis, but also its situation background, genre, intonations, amount of speakers/listeners, various extralinguistic aspects of their perception, and many other factors. This approach when formalized and applied in the tasks of Lexical and Semantic modelling, Text analysis, Machine translation, and other applications will give us an opportunity to refine all the above mentioned operations with natural language structures.

Formalization of different language processes and units is downside of Lexical and Semantic modelling problems. Such thorough language analysis requires adequate formalisms to fix all relations between language concepts in a complete and accurate way. Recognized expert in the field of language knowledge representation and acquisition, semantic relations modelling, and also of Semantic Web and language ontologies design is John Sowa [16].

He developed new methods of logic and ontology application in the systems of language understanding and relevant logic reasoning [17, 18, 19]. Language of conceptual graphs description which he designed as well was accepted as one of the principal dialects of the standard ISO/IEC [20].

Conceptual graphs (CGs) are a system of based on the existential graphs of Charles Sanders Peirce and the semantic networks of artificial intelligence [21].With a direct mapping to language, conceptual graphs serve as an intermediate language for translating computer-oriented formalisms to and from natural languages. CGs have been implemented in a variety of projects for information retrieval, database design, expert systems, and natural language processing (www.jfsowa.com/cg/index.htm). Below are some examples of natural language structures formalization using John Sowa's CGs (Figure 2, 3):

Figure 2. Conceptual graphs built for the expression «Every cat is on a mat».

In KIF language (Knowledge Interchange Format), whose author is also John Sowa, the expression visualized at Figure 2 could be written as follows:

Such representation form is convenient for natural language structures representation.

(forall ((?x Cat)) (exists ((?y Mat)) (On ?x ?y))) (2)

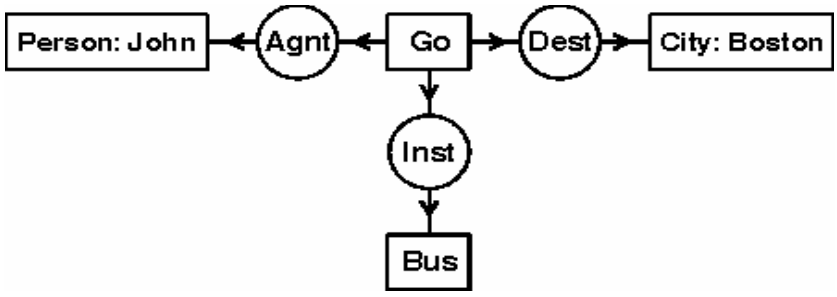


Figure 3. Conceptual graphs built for the expression «John goes to Boston by bus».

3 Additions to existing approaches of Lexical and Semantic modelling

In the previous paragraph some of principal methods and approaches to Lexical and Semantic modelling and its applications have been overviewed.

In general, while studying existing techniques developed to produce thorough analysis and parsing of natural language lexical structures, the following basic assumptions of Lexical semantics and its applications have been revealed:

- Usage of the concept "Semantic category" appeared to be rather fruitful when it came to describing lexical groups and their interrelations (particularly, in special problem-oriented dictionaries). Lexemes belonging to a certain category are marked by special descriptors [1];
- Lexeme descriptors, their attributes and relations proved their effectiveness when divided into two subsets. The first one is meant for describing semantic characteristics like ARTEFACT, SUBSTANCE, SITUATION, INFORMATION, etc. The second one – for describing binary semantic relations as PART(A,B), FORM(A,B), BELONG(A,B), etc. [1];
- Generic and specific in lexical meaning combined by logical operators allow of modelling the component semantic word structure [1, 2];
- Electronic semantic dictionaries are easy-to-use when they have data base architecture. This gives an opportunity to fix and formalize basic concepts, their attributes and relations[1, 7];
- Application of system of patterns minimizes mistakes of manual input of new concepts and associated structures. It also contributes to accurate Lexical and Semantic modelling of the natural language structures [2];
- Every word meaning considered should be represented in the form promoting a clear and complete explanation of the "language behavior" of a word in this particular meaning [4, 5];
- It is necessary to demonstrate in what way word meanings are associated with each other, i.e. to build a word meanings hierarchy or even a paradigm of its meanings general for a certain category of words [4, 5];
- The concept "Semantic role" is vital to words meanings descriptions, their links to language reality and reality in general. On one side, this furthers exposing those invariant components in semantics of a word that remain when it's deriving new forms. On the other, it allows of detecting the communicative shifts characteristic of various transformations not revealed before [4];
- The Rule of Semantic Coordination should be taken into account when design and development of new Lexical and Semantic methods are taking place, namely: one chooses such understanding of a sentence given which is characterized by maximal repetitiveness of the semantic elements, i.e. the meaning of a polysemic word is getting obvious from the context [7, 8];
- Meanings of language units correlate not immediately with the reality itself but with a language speaker notion about it [7, 10];
- Principle of the Integrative language description: vocabulary and grammar form a compact unity and are "adjusted" to each other [7];
- Meaning in any natural language is represented as an information structure encoded by people at the mental level [13];
- There is no need in interpretation of a conceptual structure by a structure of other level, since expressions of the conceptual structure are interpretations themselves (so, the question is in the choice of the structures formalization and representation tools) [14];
- Context and semantic primitives are not only words, phrases and even large fragments of text encircling an expression/sentence under analysis, but also its situation background, genre, intonations, amount of speakers/listeners, various extralinguistic aspects of their perception, and many other factors [10, 14].

Apparently, many methods and approaches from this list of basic assumptions of the Lexical and Semantic modelling have been further developed and applied at various stages of Semantics evolution. However, this doesn't mean that all existing techniques and approaches have been already developed, and problems of Lexical Semantics have been solved as well. Problems of lexical and structural polysemy, optimal representation of lexical and semantic structures in text parsing and analysis tasks, integration into information systems and domain-specific adjustment, and many other problems are still urgent. Hence, we would like to make some additions to the existing approaches.

As it was emphasized in the paper, it is a complicated task to develop Lexical and Semantic methods according to some abstract situation, or provide for all possible cases of language structures representation and usage. As a rule, everything depends on the particular applications, or some category of words and their specific features (say, English verbs), or semantic categories (e.g., semantic binary relations), etc. This leads us to a conclusion that in order to optimize approaches of Lexical Semantics at large, one needs to optimize some part of its range of problems.

Recently more and more efforts aim at maximal adjustment of linguistic resources to precise and complete modelling of natural language entities and processes. This implies qualitative change in design of problem-oriented information systems.

Therefore, as an approach adding new features to the existing ones, we introduce a formalism that has many fruitful applications in various domains. This is JSM-method of the automatic hypothesis generation originally elaborated by V.K. Finn [22, 23]. This method was named after John Stuart Mill, the English eminent philosopher, sociologist, economist, etc. He also was the author of some formalisms (Mill induction, rules of induction reasoning, inductive methods, and so on) [24]. These formalisms were applied in the JSM-method and were founded on the assumptions that it

is necessary to generate hypotheses about objects properties and effects they produce using partly defined data bases. The result of the method work would be confirmation either disproof of the basic hypotheses. For instance, method of resemblance of Mill's induction is formulated as follows: resemblance of facts implies presence/absence of an effect and its repetitiveness [24].

The main idea of the method is related to formalization of cognitive procedures interaction, namely, of induction, analogy, and abduction. This interaction coordinates Mill's ideas about induction with Peirce's abduction [25], Popper's claim to falsify generated hypotheses [26] and plausible reasoning for knowledge discovery according to Polya [27].

JSM-based systems involve instruments that may be applied in various fields of science in which knowledge is weakly formalized, data is structured, and data base contains both positive and negative examples of some objects. But JSM-method proved its particular effectiveness in machine learning. It was corroborated in the task of semantic dictionary completion [28]. However, as semantic dictionary and associated intellectual system used for testing JSM-method approaches were small-scale prototypes, and data array was rather limited, one can't judge about the method's effectiveness for Computational linguistics in general.

So the given below propositions implying JSM-method are considered only from the perspective of the task of Lexical and Semantic modelling of the Information Monitoring system.

In general case, the Information Monitoring system is designed to monitor, analyze, and evaluate some domain/system/program, etc. using some data integrated into the system or external sources as a base for necessary calculations and assessment.

Since plenty of different specialists are involved in the monitoring process (e.g., subject domain experts, evaluators, IT-specialists, decision-makers, etc.), coordination of indicators[1)] meanings and their understanding by specialists becomes problematic. Moreover, indicators are characterized by novelty and weak associations between indicators names and concepts they are designating. Taking toponyms as analogous examples which names are rarely associated with geographic position they are attached to, one can hardly grasp an indicator meaning from its name. For instance, users of the Information Monitoring system consider "Performance indicator" and "Effectiveness indicator" equivalent. But to understand meanings of these indicators when working with the system, one needs to be aware of the algorithms of their calculation first [29].

Sometimes substantial information about some indicators might be found in regulations and standards, or scientific papers but it's unpredictable and rare.

All these aspects affect the final result of the Information monitoring system performance and decision-making. That is, one of the dramatic factors of its successful performance is unambiguous coordinated indicators names and their explicit definitions.

This involved the necessity in terms dictionary design which would be integrated into the system and take into account all mentioned specific features of the subject domain, thus facilitating monitoring and evaluation procedures. The dictionary was called Semantic dictionary, as the system required coordinated meanings explanations of its terms and representation of their interrelations [30].

Thus, methods and approaches of Lexical and Semantic modelling gained significance in the field of Monitoring and associated information systems.

As it was noted at the beginning of the section, to accomplish semantic analysis one feels necessity in applying the Rule of Semantic Coordination which in its brief form states that the meaning of a polysemic word (one may be interested in) is getting obvious from the context [7, 9].

In JSM-formalism a notion of context is one of the basic ones as well. For instance, if an inductive conclusion depends on some condition X expressed by a formula not equivalent of such Y and Z that correspond to cause and effect of the conclusion, such conclusion is called *context-dependant* [24]. Therefore, when analyzing some regulations or papers containing terms of monitoring and their definitions, one may apply such inductive conclusion which would take into account basic information about these terms and their context – to add new terms/concepts, their relations and other characteristics into the Semantic dictionary.

As to terms, definitions, examples, and relations already included into the dictionary, according to JSM-method they might be related to Base of facts. It is a base containing initial information for a logical conclusion. In the Base of facts this data would correspond to (+)-facts and (–)-facts that allow of detecting cause-effect relations in texts and construct hypotheses about their belonging/not belonging (i.e., (+)-hypothesis and (–)-hypothesis) to some natural language structure from Monitoring and indicators system described in the dictionary. It should be noted that descriptors of binary semantic relations might be used as initial facts (see the beginning of the section, basic assumptions of Lexical and Semantic modelling):

$$\text{BE\_INDICATOR}\ (X,\ У=\{\text{False},\ \text{True}\}),\ \text{PART\_OF\_DEFINITION}\ (Z,\ \text{Def}) \quad (3)$$

As (+)-hypotheses and (–)-hypotheses one may use components of new terms/concepts and their derivatives context (i.e., phrasal and supra-phrasal structures encircling the term sought-for). Consider an example of JSM-method approaches application and usage of the basic assumptions of Lexical and Semantic modelling from the perspective of the Semantic dictionary completion in the Information Monitoring system (Example 1).

[1)] Indicator – an index of quantitative estimate calculated on the base of information resources of a Monitoring system.

Example 1

(+)-facts
Performance indicator
Index of the RAS program effectiveness
Peer review of RAS Institutes activities

(–)-facts
Effectiveness of a medicine
Intelligence quotient
Indicator of oxidation-reduction processes

Fragments from the Information Monitoring system taken for analysis:

<…>On the base of the Decree 201 lies the law on system of result indicators of Federal programs of the RAS basic research. In order to define effectiveness of the program one should take into account the following indicators <…>

<…>Apparently, it is not enough to use results of logic-maths tests and assotiation tests as a general intelligence quotient of the senior age groups <…>

<…>accomplishing peer review of research departments of the RAS led to <…>

(+)-hypotheses
On the base of the decree lies the law
To define effectiveness
Accomplishing peer review led to

(–)-hypotheses
Apparently, it is not enough to use… as a quotient

When using such data in the context-dependant inductive conclusion of the JSM-method of the automatic hypotheses generation the Semantic dictionary is completed by terms/concepts «*Result indicators of Federal programs of the RAS basic research*» and «*Peer review of research departments of the RAS*».

Obviously, even such a small example demonstrates that there is necessity in conclusion results validation (new terms/concepts in the dictionary) and more detailed processing of the conclusion elements. But Example 1 makes it evident that formalism of the JSM-method can be adapted to the task of contextual retrieval and semantic analysis with further completion of the Semantic dictionary of the Information Monitoring system. But this is determined by initial data, limitations and requirements to the result of the linguistic resources performance.

4 References

[1] N.N. Leontjeva. Categorization of the units in the Russian all-semantic dictionary. (Леонтьева Н.Н. Категоризация единиц в русском общесемантическом словаре (РОСС) //Труды Международного семинара Диалог'98 по компьютерной лингвистике и ее приложениям. Т.2.- С.519-532). – Published in Russian.

[2] N.N. Leontjeva, S.J. Semenova. On reflecting polysemy in the automatic semantic dictionary. (Леонтьева Н.Н., Семенова С.Ю. Об отражении полисемии в прикладном семантическом словаре //Компьютерная лингвистика и интеллектуальные технологии. Труды международного семинара Диалог 2002. Протвино, 6-11 июня 2002 года. – М., "Наука", 2002. – Т.2. – С. 489-496.) – Published in Russian.

[3] N.N. Leontjeva, S.J. Semenova. The Semantic dictionary RUSLAN as an instrument of the computer comprehension. (Леонтьева Н.Н., Семенова С.Ю. Семантический словарь РУСЛАН как инструмент компьютерного понимания // Понимание в коммуникации. Материалы научно-практической конференции. М., 2003.) – Published in Russian.

[4] G.I.Kustova, J.V. Paducheva. Dictionary a lexical data base. (Кустова Г.И., Падучева Е.В. Словарь как лексическая база данных. - Вопросы языкознания, 1994, N3.) – Published in Russian.

[5] Lexicograph project site: http://lexicograf.ru/ideology/

[6] J. Kurilovich. Notions about word meaning (Курилович Е. Заметки о значении слова // в его кн.: Очерки по лингвистике, пер. с польск., англ., франц., нем. – М., 1962.) – Published in Russian.

[7] J.D. Apresyan. Selected works. (Ю.Д.Апресян. Избранные труды. Том 1. Лексическая семантика. Синонимические средства языка. М., 1995.) – Published in Russian.

[8] V.K.Gak. Semantic structure of a word as a component of the semantic structure of an utterance. (В.Г.Гак. Семантическая структура слова как компонент семантической структуры высказывания // Семантическая структура слова. Психолингвистические исследования. –М., 1971.) – Published in Russian.

[9] V.K.Gak. About semantic syntagmatics problem. (В.Г.Гак. К проблеме семантической синтагматики // Проблемы структурной лингвистики. – М ., 1972.) – Published in Russian.

[10] Wierzbicka, Anna. Semantic Primitives. – Frankfurt/M., Athenaum-Verl., 1972. – 235 p.

[11] J.D. Apresyan, I.M. Boguslavsky, L.L.Iomdin et al. Linguistic software for the system ETAP-2. (Ю. Д. Апресян, И. М. Богуславский, Л. Л. Иомдин и др. Лингвистическое обеспечение системы ЭТАП-2. – М.: Наука, 1989.) – Published in Russian.

[12] J.D. Apresyan, I.M. Boguslavsky, L.L.Iomdin et al. Linguistic processor for complex information systems. (Ю. Д. Апресян, И. М. Богуславский, Л. Л. Иомдин и др. Лингвистический процессор для сложных информационных систем. – М.: Наука, 1992.) .) – Published in Russian.

[13] Jackendoff, Ray. Semantic Interpretation in Generative Grammar. Cambridge, MA: MIT Press, 1972. – 400 pp.

[14] Jackendoff, Ray. Semantics and Cognition. Cambridge, MA: MIT Press, 1983. – 283 pp.

[15] Jackendoff, Ray. Foundations of Language: Brain, Meaning, Grammar, Evolution. Oxford: Oxford University Press, 2002. – 477 pp.

[16] Sowa, John F. Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks/Cole Publishing Co., Pacific Grove, CA, 2000.

[17] Sowa, John F. Language Games: A Foundation for Semantics and Ontology. In Game Theory and Linguistic Meaning, edited by Ahti-Veikko Pietarinen, Elsevier, 2007. – pp. 17-37.

[18] Sowa, John F. The role of logic and ontology in language and reasoning. Chapter 11 of Theory and Applications of Ontology: Philosophical Perspectives, edited by R. Poli & J. Seibt, Berlin: Springer, 2010. – pp. 231-263.

[19] Sowa, John F., Majumdar, Arun and Stewart, John. Pursuing the goal of language understanding. P. Eklund & O. Haemmerlé, eds, Proceedings of the 16th ICCS, LNAI 5113, Berlin: Springer, 2008. – pp. 21-42.

[20] Sowa, John F. Conceptual Graphs. In F. van Harmelen, V. Lifschitz, and B. Porter, eds., Handbook of Knowledge Representation, Elsevier, 2008. – pp. 213-237.

[21] Max H. Fisch. Peirce Semiotic and Pragmatism. Indiana University Press, 1986.

[22] V.K. Finn. On the role of machine learning in the intellectual systems. (Финн В.К. О роли машинного обучения в интеллектуальных системах // Научно-техническая информация, Сер. 2, № 12. — М.: ВИНИТИ, 1999. — С.1-3.) .) – Published in Russian.

[23] Artificial intelligence. Reference book (Искусственный интеллект. Справочник в 3 кн. . Под ред. Попова Э.В //М.: Радио и связь, 1990.) .) – Published in Russian.

[24] J.S. Mill. System of the syllogistic and inductive logic. (Милль Д.С. Система логики силлогистической и индуктивной: Изложение принципов доказательства в связи с методами научного исследования. Пер. с англ./ Предисловие и приложения В.К. Финна. – М.: ЛЕНАНД, 2011. – 832 с.) – Published in Russian.

[25] Peirce C.S. Abduction and induction. Philosophical Writings of Peirce. Ed. Buchler. Dover Publications NY, pp. 150-156.

[26] Popper, K. R. Objective Knowledge, corrected edition, Oxford, Oxford University Press (1973).

[27] G. Polya. Mathematica and plausible reasoning. (Пойа Д. Математика и правдоподобные рассуждения. – М.: Книжный дом «Либроком»/URSS, 2010.) – Published in Russian.

[28] V.K.Finn, D.V.Vinogradov, O.S. Kozhunova. Intellectual system of semantic dictionaries completion. (Финн В.К., Виноградов Д.В, Кожунова О.С. Интеллектуальная система пополнения семантических словарей // Программные продукты и системы, № 2, 2006. – с.27-30.) – Published in Russian.

[29] I.M. Zatsman, O.S. Kozhunova. Semantic dictionary of the Information Monitoring system in the field of science: tasks and functions. (Зацман И.М., Кожунова О.С. Семантический словарь системы информационного мониторинга в сфере науки: задачи и функции. // Системы и средства информатики. Вып. 17.- М.: Наука, 2007.- С. 124-141.) .) – Published in Russian.

[30] Olga S. Kozhunova. Semantic Dictionary: heading toward ontology of the Information Monitoring System // Proceedings of WorldComp'10, Las Vegas, 2010. Pp. 934-941.

SESSION

XI TECHNICAL SESSION ON APPLICATIONS OF ADVANCED AI TECHNIQUES TO INFORMATION MANAGEMENT FOR SOLVING COMPANY-RELATED PROBLEMS

Chair(s)

Dr. David de la Fuente
Dr. Jose A. Olivas

Integrated System for Managing Traceability in Intermodal Logistic Environments

Durán, J.[1]; del Pino, B.[1]; Andrés, C.[1]; Ochoa de Olano, E.[2]; Diaz, M.[2]; García, N.[2]
[1] Centro Experimental del Conocimiento, Valladolid, Spain
[2] EPIG, University of Oviedo, Gijón, Spain

Abstract - *The interest in having traceability systems covering all the stages of the intermodal logistic chain at the trading partners disposal has increased in the recent years, in order to reduce business risks, achieve greater efficiency, support product safety, and provide information to consumers and partners. In this paper we address the subject of managing traceability features in the context of supplying goods, which is part of the research carried out under the project SITIM. Firstly we present an overview that describes the problems connected with the intermodal logistics area and analyzes the general aspects of traceability systems. Then, we present the particular case of a logistic operator that provides car transport logistics from a point of loading until its final destination, followed by the description of the characteristics of the information system we propose for the traceability of cars through the stages of the freight transport.*

Keywords: intermodal logistics, supply chain traceability, traceability management systems, car distribution

1 Introduction

Creating and managing a supply chain in a logistic intermodal environment for the delivery of a product from the manufacturer to the end consumer is a difficult task, as it involves the interaction of numerous business partners and decision makers who need to work in a coordinated manner to improve the overall functioning of the chain, but each one imposes constraints and targets cost effectiveness. The complexity grows when considering intermodal transportation, which consists of the combination of at least two different transport modes, such as rail, river, sea or road, in a single logistic chain, without any transfer of goods between containers. As a result, developing intermodal logistics systems that can build and operate logistic chains according to business and regulation policies, facilitating the actions to be carried out by the various agents involved in the process, has attracted the attention of research communities.

In recent years, operational research [1] applies to various aspects of logistics, such as management of freight and delivery systems, or management of flows of materials and products within the supply chain from the estimated demand of final products. However, they usually focus on transportation issues involving a single mode. Another aspect which is taking increasing awareness among the users and the Administrations is the development of logistics safety and reliability, both of which directly influence the internal points of the supply chain and the final client. With regard to the client, the improvement in the levels of reliability allows the assurance of the availability of the products requested, in the desired conditions, on the dates planned and the fulfillment of all necessary information requirements. But this will not be achieved successfully if the product does not go through sufficiently safe processes, during the supply chain, for which these two concepts are directly related. Due to the importance of ensuring product safety and quality, traceability has become an important feature that business solutions should take in consideration.

Traceability can be considered the key concept around which the contents of this paper turns, which presents part of the research carried out under the project SITIM, "Analysis, development and evaluation of Intelligent Transport Systems in an intermodal freight environment", supported by Spanish Ministry of Publics Works and being developed since January 2009 under the Spanish Scientific Research, Development and Innovation National Plan. Its final objective is the analysis, development and evaluation of the use of intelligent transport systems in freight transport in an intermodal environment, in order to improve effectiveness and sustainability through the use of information and communication technologies combined with heuristic techniques for process optimization.

The purpose of this paper is to present the broad outlines of our research tasks, for the development of an information system for the traceability of goods through the stages of the freight transport from the point of loading until its final destination at the customer. The requirements of the problem are particularized for the real case of one of the integrants of SITIM project, whose business consists of the transportation of vehicles from an origin point to the concessionary destination.

The rest of the paper is organized as follows. Section 2 introduces some concepts about intermodal logistics and traceability. Section 3 analyzes the characteristics of traceability systems. Section 4 presents the real case under study. Section 5 describes the background and needs of the customer. Section 6 presents the description of the solution proposed. The paper ends with some conclusions.

2 Intermodal logistics and traceability

At present, the development of logistics systems that can build and operate logistic chains according to business and regulation policies has taken much interest among the research communities. The task of creating and managing a supply chain is not easy because the delivery of a product from the manufacturer to the end consumer involves the interaction of numerous business partners, agencies, councils and organizations, which imposes constraints and is targeting cost effectiveness [2]. The complexity increases when considering logistic chains in which transport is performed in an intermodal environment, that is, it combines at least two successive modes in a single chain, without a transfer of goods between containers. Most of the route runs on rail, sea or river, to the detriment of road transport which is usually limited to the end of the intermodal chain.

The large number of decision makers that involves intermodal transportation, who need to work in a coordinated manner to improve the overall operation of the chain, makes it necessary to use tools for decision support to facilitate the actions to be carried out by each of them. The research community, conscious of the growing importance of intermodal freight logistics, is studying, analyzing and developing intelligent transport systems with the objective of improving the effectiveness and sustainability of the intermodal supply chains though the use of information and communication technologies. These systems are usually combined with *metaheuristic techniques* [3], which allow to optimize the processes carried out daily in intermodal logistics and to solve the complex problems that appear primarily because of the large search space and number of possible solutions, making it usually impossible to explore it thoroughly to find the optimal solution with current technology.

The problems related to the area of intermodal transport logistics are categorized in [4] into several groups: supply chain management; supply logistics; planning and control of production and operations; storage, packaging and handling; management of transport and mobility; distribution logistics; and reverse logistics. The study addresses the use of metaheuristic techniques, especially *genetic algorithms*, for general optimization of some of the main problems identified, such as problems with routing and distribution, storage problems, design issues of the supply chain, problems in reverse logistics and optimal network design problems. This problems and the application of algorithm solutions for general optimization are widely studied in the literature [5] [6] [7] [8] [9] [10] [11] paying much attention to genetic algorithms.

In addition to the efforts of the logistic companies to optimize the processes involved in the multimodal transport of goods, it has been added in recent years the interest of businesses, consumers and authorities in establishing systems to trace the history of products in the supply chain, mainly for food [15], pharmaceutical products and beverage, but also in other industry sectors such as manufacturing and fashion. This way, the establishment of a product traceability solution covering all the stages of the multimodal transport chain can be used by trading partners in a supply chain to reduce business risks about legal compliance, to achieve a greater degree of precision and efficiency in product recall or withdrawal, and to comply with specifications of each partner [12]. It also needs to be considered for an efficient logistics and an effective quality management, as well as for supporting product safety, providing information to consumers and partners.

3 Traceability systems description

The traceability systems, technologies and standards developed to carry out traceability vary from large to small enterprises [13]. The first ones are characterized by a tightly aligned supply chain, typically supported by a considerable use of information and communication technology. Usually, they already use a traceability system, which is typically very efficient and fully automated. On the contrary, small enterprises implement traceability rarely and when they do, they add the traceability management to their normal operation often in a manual or semiautomatic way, decreasing efficiency and increasing costs.

The complex task of building a traceability system involving all the stages of production, processing and distribution is under investigation by scientific and industrial bodies. The study of Cimino [12] proposes an organic approach to manage the related aspects, such as traceability semantics, scalability information management and traceable items (*lot*) identification. Traceability is defined as the ability to trace the history, application or location of an entity by means of recorded identifications through the whole or a part of a production chain from raw materials through transport, storage, processing, distribution and sales (*chain traceability*) or internally in one of the steps of the chain (*internal traceability*). Each lot must be both traceable and trackable, where *tracing* means to identify its origin by *tracing* back in the supply chain, whereas tracking means to follow the path of the product through the supply chain from supplier to consumers.

3.1 Identification techniques

The first step towards an effective traceability information management is *lot identification*. Automatic identification technologies help industry and commerce to reduce labor costs and stocks, shorten production times, improve customer service and ultimately increase productivity and profits. Zhang [14] remarked a description of the main technical solutions available:

- *Alphanumerical code*. It consists of a sequence of digits of various sizes placed on products or on its packaging that provides a univocal key to retrieve traceability data stored elsewhere. They are suitable in many kinds of industries, simplifying and

reducing the cost of the design phase. But the performance is not particularly good since code writing and reading are not automatic, and the risk of damaging data integrity is high.

- *Barcode*. It consists of an electronic readable tag placed on the products or on its packaging that represents an identifying number and allows obtaining related information of the item from a database keyed to it when detected and identified by the reader. It is less attractive for production processes and distribution because of its usually generous dimensions and ease to be damaged, and because the system requires human intervention for the scanning process and so there is room for error and inefficiency.

- *RFID tag*. It consists of a chip that stores a serial number and other information attached onto or implanted into any surface of an item. The chip contains an antenna that transmits the identification information via radio waves to a reader. The reader converts them into digital information that can be passed to an enterprise information system.

Given the fact that each technique has its own advantages and disadvantages, the selection of a particular technology depends on the nature of the product and the stage of production [15], although it is also possible to mix several techniques for modeling traceability information throughout the chain, as described by Zhang [14]. Effective and secure communication methodologies should be adopted in order to ensure accurate and secure exchange of the information recorded on various types of codes or IDs along the stages in the production process and among various stakeholders in the supply chain.

3.2 Actors involved

A recent study [2] shows a system that captures the entities and relations involved in the process of creating and using a logistic chain and managing traceability data. It identifies a close relationship between the logistic chain and the business partners involved, who are classified in:

- *Primary business partners*: They include manufacturer, carrier, warehouse and retailer, and perform one or more processes that affect the lots, such as integration (mixing, packing), division (cutting, splitting), alteration (heating, freezing, drying), movement, acquisition, providing, manufacturing, transport, storage or selling.

- *Secondary business partners*: They include the authorities, who verify compliance with safety regulations and take actions when required, and end consumers, who can query for information about a product they have bought from retailers.

When talking about managing freight traceability in a multimodal transportation context, other distributions appear in the literature. The system in [16] shows how the actors implied are distributed in three categories according to their responsibility: shipper (salesperson or purchaser), organizers of transport (transport commissioner, transit agent, etc.) and logistic operators (transport, managers of platform, ship-owners).

3.3 Traceability data

The model described in [2] divides the traceability data that describe a traceable item into three kinds:

- *Manufacturing data*. It provides information about the product specifications, such as manufacturing and expiration date, storing conditions, etc, which does not change throughout the logistic chain.

- *Local processing data*. It describes the processes that suffer the traceable item inside the infrastructure of a primary business partner. It includes structural data, which captures the changes in the encapsulation level of the traceable item, and specific data, that describes its manipulation through the procedures that affect it within the premises of a business partner. It is possible to implement a specified interface to expose this data to other business partners.

- *Transfer data*. It contains information about the logistic units' transfers between primary business partners, making it possible to find the sequence of a certain traceable item.

3.4 Communication platform

It is necessary to establish a module for the exchange of messages between the different actors implied to ensure a suitable performance of the traceability system. It also feeds the database with up-to-date information related to the evolution of the products through the intermodal transport chain.

The objective is that the platform satisfies the needs of interoperability, implementation and mobility of the informational solutions used for goods transportation, for which most of the research communities agree to propose solutions according to SOA (Service Oriented Architecture) architecture by means of web-based services [16] [17].

The basic architecture of a web service is based on SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language) and UDDI (Universal Description, Discovery and Integration). Indeed, a web service is described by using WSDL and is registered in UDDI registries to facilitate research (discovery) thereafter. The task of transport and exchange of data between systems is done through SOAP protocol.

4 Case study

Current research is carried out under the project SITIM, "Analysis, development and evaluation of Intelligent Transport Systems in an intermodal freight environment", supported by Spanish Ministry of Public Works and whose main objective is the application of intelligent transport systems for freight transport in an intermodal environment, in order to improve effectiveness and sustainability and increase visibility along the whole logistics chain.

One of SITIM participants, the company Bergé y Cía (from now Bergé), is one of the major Spanish corporate groups and has been the basis of the system assessment and implementation. It has a sizeable international projection in sectors such as ports, car distribution, logistics, renewable energy and finance.

The automotive industry is the sector assessed by the present research, in which Bergé is nowadays the mayor private car distributor in the Iberian Peninsula and Latin America. It represents a total of 28 brands (between distribution and retail trade) in seven European (Spain, Portugal and Poland) and Latin American (Chile, Peru, Argentina and Mexico) countries. Moreover, it collaborates with an independent network of more than 400 car dealers. The system has been oriented to this sector, including real characteristics of the processes followed by the vehicles and actors involved in order to reach realistic and practical results.

To simplify the design requirements, the scope of the study is limited to the case of car distribution in Spain, where Bergé works primarily in a single mode transport environment, providing the car transport logistics from a point of loading, such as a car warehouse or manufacturer, until its final destination, that can be an intermediate warehouse or a car dealer. Intermodal is also present as some loading and unloading stations are placed on maritime ports or rail nodes.

5 Background. Customer care-abouts and needs.

In their condition of big logistic operator, Bergé has thousands of customers and business partners scattered across a wide region. There is a long list of customer care-abouts that can be identified through the full stream of operations, which includes administrative issues, commercial relationships, giving enough volume to the transportation subcontractors net, handling goods effects upon persons, handling effects (such as knock, damage, etc.) upon the goods themselves, lot of sources of inefficiencies like the use of transportation capacity, waste due to inefficient routes, the waiting time in loading and unloading nodes, and so forth [4].

This work focuses on the transactional issue of keeping the customers properly informed about the state of their goods (cars in this case) when customers make potentially critical decisions. A classical approach to tackle this problem is to regularly (i.e. each morning) issue pieces of information to be sent to customers, but it involves some severe shortcomings. Just to mention one, it is that news about changes of the state of the cars in the overall distribution system happens every minute, and the regular distribution cannot be done with such a frequency.

On the other hand, there are two different views about traceability of vehicles: one refers to customers and the other one fulfills traffic operator needs.

External traceability aims to cover customer needs in terms of '*where is my transportation order right now?*', '*is there anything important I need to know about it?*', '*what is going to happen with it next?*' and '*when I will have my order completed?*' Details offered to customer should be different depending on the type of customer. OEMs[1] need more details, exception reports, key descriptive statistics about the service and enhanced navigability across data. On the other hand, retailers need focus on particular facts about a specific order they are interested in tracking it at any moment of time.

No matter what the customer type is, traceability must offer agile exploring services to help customers to raise the important questions. The vision of caring about customer needs that real time traceability brings to the patrons is a step ahead in promoting continuous improvement inside the organization. It is a very well known and demonstrated fact that the principle of being VISIBLE is a core key principle to foster an attitude towards excellence [18].

Once we have described the external traceability, focused on customers, the internal traceability is tackled, whose users are internal employees in Bergé (traffic operators). A significant problem of a traffic operator is to have real time information about fleet units. SITIM has brought to the system the advantages of ubiquitous technology, replacing the traditional reporting in order to collect information about each item of the fleet by delegating to each transportation unit the responsibility to update key valuable information.

As the equation stated in Georges Eckes' book '*Q*A = excellence*' [19], in order to move ahead in the path of excellence we should deal not only with technical issues, but also have into account and manage the cultural side of changes. That is to say, making technology accepted by key players. SITIM seeks functionality as well as simplicity, interoperability and usability. Special tests have been used: *five second test* to see whether the user can get the key message without any effort; *card sorting* techniques, to arrange the information displayed in an intuitive way; and the *tree Jack test*, to verify if those tasks can be done in a time frame by most of the users.

1 OEM : original equipment manufacturer

6 System implementation

6.1 Proposed system schema

The project team devised the system characteristics in several dimensions. One is about system architecture. The first consideration was to implement external traceability upon the available system in Bergé - the ALTRA[2] application. This option was discarded because the goal was to provide a simple mechanism to make the system accessible for an extensive group of customers. The decision was taken by comparing the assessed scoring done by the team through the decision matrix that is showed in Table 1.

Table 1. System architecture decision matrix.

Factor	*Description of the factor*	*Weight*	*ALTRA*	*Web service*
Ubiquity	*The application should be accessible to many external users*	*9*	*1*	*2*
Service level	*External users and internal users of ALTRA can work without interferences*	*8*	*1*	*2*
Easy to maintain	*The lower the number of system components, the better to administrate*	*5*	*2*	*1*
Functionality	*The effects of the platform in order to provide services for users*	*10*	*1*	*2*
Extras	*Facility to add extra services, like graphics, corporate image, etc*	*6*	*1*	*2*
Security	*The advantages in terms of prevent inappropriate access to data*	*7*	*1.5*	*1.5*
	Total SCORE		*53.5*	*81.5*

The final decision was to use a web service approach with a series of connectors to be continuously ready to accept telegrams coming from ALTRA. When data is updated in ALTRA, the stubs (pieces of software that detect changes in ALTRA database) send telegrams to be captured by SITIM web services.

The general schema for the proposed platform is shown in Figure 1. It includes the connectors with ALTRA and the communication approach to collect relevant data from the fleet (truck drivers).

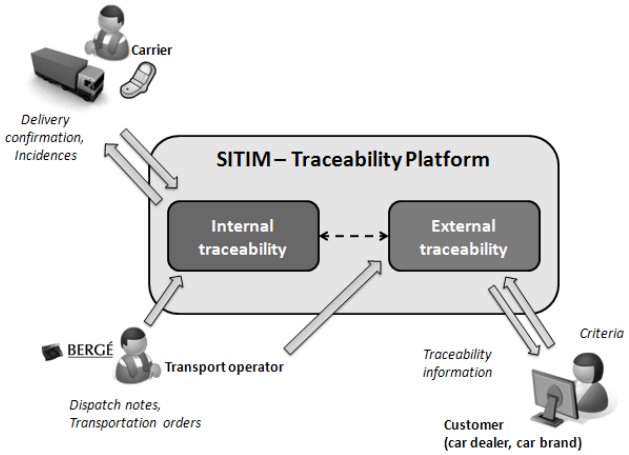


Figure 1. General schema for the proposed traceability system

[2] ALTRA: Application used in Bergé to manage transportation orders.

It is important to remark that the same contents, in terms of semantics, are presented under different formats to external users (mainly interested in knowing where their cars are or when they can expect to have the required transport service completed) and internal users (more interested in receiving alarms and help from the system when they have to deal with problems).

6.2 Functional description

The functionality of the system must provide users with solutions to customer needs and customer care-abouts. Figure 2 shows a structured view of the traceability application contents, described in the next paragraphs.

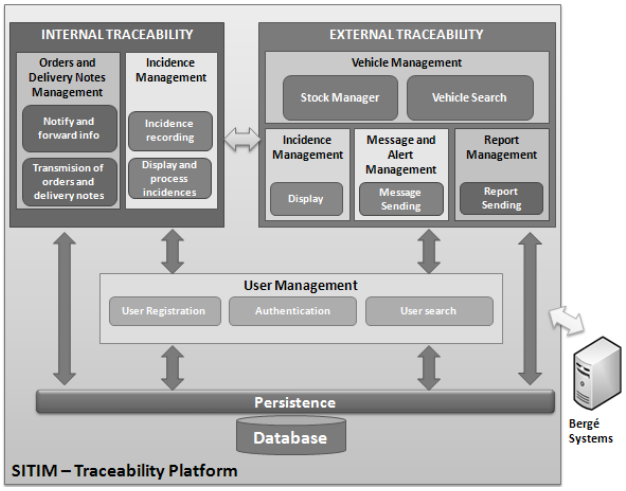


Figure 2. Functional description for the proposed traceability system.

6.3 External traceability

This module runs on a web platform in order to be ready at any time for any customer to answer critical questions. Questions about what is happening to their transportation orders, which generally travel through a series of logistic segments, and what can they expect afterwards. The business value of this service is to promote the corporate reputation of Bergé by reducing anxiety and uncertainty among clients, as well as reducing cost due to the time spent in attending customer calls. This module includes the following functionalities:

- *Items of transportation orders*: Customers use this module to browse transportation orders. Customers may be interested in collecting details from orders related to a certain car model to be received at a particular arrival point or within a zone, in a range of dates, or they can track a chassis number, etc

- *Interoperability services*. This module allows customers to export files in several formats (PDF, Excel, etc.) with the purpose of simplifying the chunk tasks that customers have to face in order to move data across different platforms. Reporting facility is also provided so that it can be easily sent to different destinations.

- *Troubleshooting*: Any issue with synchronization of data is trapped and thrown to this module. Typical reaction is to reveal enough evidences so that the user can work with reduced information until the support agent deals with the causes of the problem, puts appropriate countermeasures and drive the system back to its normal situation in terms of data quality.

- *Managing security*. The crew of commercial agents is provided with tools to distribute the workload of taking care of external customers (providing them their accounts, security and help).

6.4 Internal traceability

The business value of internal traceability can be described in several aspects. One is enhancing customer perception of Bergé's service when dealing with trouble. Another is to manage traffic operator issues. A standard procedure for traffic operators to avoid usual problems is implemented by making the system collect news with the lowest manageable delay so that alarms are raised with a very low rate of error type I and error type II[3].

The messaging system has to be attractive enough to make people use the system –as it makes their work easier in terms of communicating where the transportation unit is, getting at every stage of their route– instead of keeping out of it.

Other important point to mention is to provide services linked to the *business analytics* concept. Gaining continuously competitive edge means to understand customers every day better and better. To do so, it is important for traffic operators to get statistics of the quality of service (QoS) and to manage a balanced scoreboard and a PDCA[4] cycle for improvement.

Of course, there are also savings in the total time spent by operators in attending customer phone calls. There is nothing wrong in having good communications with customers, but this is not the main point. We aim that such communication is addressed to develop business rather than having to calm down and to provide compensation to retain an upset customer. A brief summary of the internal traceability functionalities is:

- *Dealing with key events in the transportation route*: this module collects the key events (loads and unloads, for instance, as well as truck breakdowns and other sort of disruptions in the process).

- *Managing deviations*: similar services are provided to take care of unplanned events (breakdowns, wrong delivery, safety issues, etc) so that the system has a detailed view in real time of the general situation of the orders and the fleet and Berge's traffic operators are noticed to deploy appropriate actions.

- *Managing additional user profiles*: traffic operators and truck drivers are included.

6.5 Some technology considerations

To communicate the fleet (outsourced service) with Bergé's system, the selected option is SMS messaging through cellular phones. The supporting arguments for this decision among other alternatives (web application, offline, etc.) were the advantages of simplicity when making it available to the general population of the fleet, and the availability to use the system without time frame restrictions, without a difficult learning curve to train people to use the system, and the fact that it would not increase the cost for the subcontractors in the phone bill compared to what they are currently spending.

The kind of web service used in this solution is based upon the interaction between the module that is sending or receiving the SMS, and the SMS provider. This approach decouples the application from the communications with the fleet, while keeping the integration of data.

Some relevant features of this services that are important for the quality of the system are:

- Multiple SMS message types (in terms of contents) can get associated to different threads depending on key words.

- Sending messages to the fleet can be done in an automatic scheduled way triggered by clock events or other sort of events.

7 Conclusions

The Traceability Platform SITIM provides a complete variety of functionalities that appear in a vehicle transportation chain. It has been implemented in the context of the main international car distributor Bergé. Specifically, SITIM has been implemented for Spain in a single mode transportation environment, although intermodality general issues have also been considered. Recent technologies and software platforms have proved to be adequate solutions to tackle some problems that were partially addressed some years ago. For the software implementation, the agile methodology called Scrum has been successfully used with remarkable quality results.

SITIM traceability feature has been designed in two parts. The 'External traceability' part is designed for the final costumers (car dealers) and Bergé transport operators,

[3] Errors I, II: In the statistics domain, an error of type I -also known as α error- it means to raise an alarm when there is nothing wrong, and an error of type II -also known as β error- consists on not raising an alarm when there is a system issue.

4 PDCA: Deming's Plan, Do, Check, Act cycle to implement continuous improvement in a process.

providing full vehicle management as well as some modules to create, control and process incidences, messages and reports. On the other hand, the 'Internal traceability' part is used for the fleet drivers and Bergé traffic operators and offers a way for dealing with key events in the transportation route and managing deviations.

Some future improvements could be proposed to enhance the functionalities offered by SITIM with other recent technologies and that now are being considered. Firstly, using effective and inexpensive techniques to use a 3D (or even 4D) approach to tackle traceability (for certain products). Other relevant aspect is to develop additional features in the field of business analytics for the main agents to cooperate in making decisions, better communicating, etc. in a highly dynamic environment.

8 References

[1] C. Macharis, Y. M. Bontekoning. "Opportunities for OR in intermodal freight transport research: A review". European Journal of Operational Research, vol. 153, issue 2, pp. 400-416, March 2001

[2] I. Salomie, M. Dinsoreanu, C.B. Pop, S.L. Suciu. "Logistic chain generation with traceability features using web services composition". Proceedings of IEEE International Conference on Automation, Quality and Testing, Robotics, 2008 (AQTR '08), Theta 16th edition, vol. I, pp. 393-397

[3] M. C. Vélez, J. A. Motoya: "Metaheurísticos: Una alternativa para la solución de problemas combinatorios en Administración de Operaciones". Revista EIA, ISSN 1794-1237, vol. 8, pp. 99-115, December 2007

[4] J. Duran, E. Abril, P. Lopez, J. Puente, J. Parreno. "Using Genetic Algorithms to Optimize an Integrated Management System for Intermodal Logistics". Proceedings of International Conference on Artificial Intelligence (ICAI '10).

[5] B. Eksioglu, A. Volkan Vural, A. Reisman: "The vehicle routing problem: A taxonomic review". Computers & Industrial Engineering, vol. 57, issue 4, pp. 1472-1483, November 2009

[6] I. Gribkovskaia, O. Halskau, G. Laporte: "General solutions to the single vehicle routing problem with pickups and deliveries". European Journal of Operational Research, vol. 180, issue 2, pp. 568-584, July 2007

[7] M. Bazzazi, N. Safaei, N. Javadian: "A genetic algorithm to solve the storage space allocation problem in a container terminal". Computers & Industrial Engineering, vol. 56, issue 1, pp. 44-52, February 2009

[8] C.A. Silva, J.M.C. Sousa, T. Runkler, R. Palm: "Soft computing optimization methods applied to logistic processes". International Journal of Approximate Reasoning, vol. 40, issue 3, pp. 280-301, November 2005

[9] J. E. Lee, M. Gen, K. G. Rhee: "Network model and optimization of reverse logistics by hybrid genetic algorithm". Computers & Industrial Engineering, vol. 56, issue 3, pp. 951-964, April 2009

[10] C.K.M. Lee, T.M. Chan: "Development of RFID-based Reverse Logistics System". Expert Systems with Applications, Vol. 36 (5), pp. 9299- 9307, July 2009

[11] A. Sirikijpanichkul, K. H Van Dam, L. Ferreira, Z. Lukszo: "Optimizing the Location of Intermodal Freight Hubs An Overview of the Agent Based Modelling". Journal of Transportation Systems Engineering and Information Technology, vol.7 (4), pp. 71-81, August 2007

[12] M.G.C.A. Cimino, F. Marcelloni. "Autonomic tracing of production processes with mobile and agent-based computing". Information Sciences, vol. 181, issue 5, pp. 935-953, March 2011

[13] A. Bechini, M.G.C.A. Cimino, F. Marcelloni, A. Tomasi. " Patterns and technologies for enabling supply chain traceability through collaborative e-business". Information and Software Technology, vol. 50, issue 4, pp. 342-359, March 2008

[14] J. Zhang, P. Feng, Z. Wu, and D. Yu. "Automatic Identification-Enabled Traceability in Supply Chain Management," in 4th International Conference on Wireless Communications, Networking and Mobile Computing, 2008 (WiCOM '08), pp. 1-4

[15] G.G.D. Nishantha, M.K. Wanniarachchige, S.N. Jehan. "A pragmatic approach to traceability in food supply chains," 12th International Conference on Advanced Communication Technology, 2010 (ICACT '10), vol.2, pp.1445-1450

[16] S. Bendriss, A. Benabdelhafid, J. Boukachour. "Information system for freight traceability management in a multimodal transportation context", IEEE/ACS International Conference on Computer Systems and Applications, 2009 (AICCSA '09), pp.869-873

[17] S. Bendriss, A. Benabdelhafid, J. Boukachour. " Data Modeling and Integration Approach for Enabling Multimodal Goods Traceability" Proceedings of the 2nd International Conference on Logistics and Transport, 2010 (ICLT '10), pp.32

[18] J. Liker, D. Meier. "The Toyota Way Fieldbook". McGraw-Hill, ISBN 978-00-714-4893-2, pp. 476, 2006

[19] G. Eckes. "Making Six Sigma Last: Managing the Balance Between Cultural and Technical Change (Six Sigma Research Institute Series)". Wiley, ISBN 978-0471415480, pp. 1-16, 2001

Heuristic Solutions to the Vehicle Routing Problem with Capacity Constraints

Pino, R.[1]; Villanueva, V.[1]; Martinez, C.[1]; Lozano, J.[1]; Del Pino, B.[2]; Andrés, C. [2]

[1] EPIG, University of Oviedo, Gijón, Spain
[2] Centro Experimental del Conocimiento, Valladolid, Spain

Abstract – *The purpose of this paper is to conduct a review in the scientific literature to know the state of the art relating to a real optimization problem. This problem issues from a research project which studies the application of Operations Research methods for the optimization and upgrading of Intermodal Transportation Systems. Concretely, the problem can be classified as a VRP type. The most suitable problems and solutions were studied, but also those which would contribute to have an outlook on how to solve this kind of problems. The current document has been divided into two main groups; exact algorithms and approximated methods. In each group, the different algorithms studied were presented. Each algorithm studied were briefly explained and mentioned the where it has been applied for.*

Keywords: Vehicle Routing Optimization, Artificial Intelligence, Heuristics

1 Introduction

The search is intended to discover the techniques and methods that are investigated and applied in the scientific world, and also to find out what tools would be useful for our specific problem.

The proposed problem concerns a logistics company dedicated to the vehicle distribution by road (road transport). This distribution is done based on orders that are received in depots spread throughout the Spanish geography. Resultant freights are loaded into available trucks according to a series of objectives and constraints, and then sent to the corresponding destination. Once these trucks have finished their current task they will be available for a new delivery. Therefore, the trucks do not follow a delivering centralized scheme, otherwise, they travel around the whole geography picking up and delivering loads.

2 Proposed objectives

Business objectives are focused on finding a balance between profitability and quality of service offered. Therefore, optimization techniques pursue this aim by using scientific and technical tools from a wide range of fields: Mathematics, Computer Science, Telecommunications, Industrial Engineering, etc. This combination of techniques allows to transform and improve the complex business reality, where the needs and objectives are not subject to a single branch of knowledge.

The main pursued aims can be summarized in: Maximize truck load factor, Minimize empty kilometers, avoid geographical spread and meet delivery deadlines.

As mentioned above, the listed goals are focused on finding the greatest efficiency in fleet management so as trucks would be permanently picking up and delivering orders, avoiding periods of inactivity. In order to increase such efficiency, it is also contemplated the possibility of collecting a vehicle in the course of a route.

3 Current operational

Company's daily operations are currently performed by a set of operators, each of one is responsible for a specific geographical zone which can contain one or several depots. The operational consists in serving the orders that are received at the depots.

Therefore, the operator's job is to plan what vehicles should be loaded onto each available truck so as to satisfy the conditions imposed and ensure that the final destination of the path allows the truck to be reloaded in a near point.

At this point, it should be noticed that a proper routing is crucial because the correct number and type of trucks must be available at the depot or reload point. On the other hand, operators extract information from their experience and from historical data to guess where there will be more order requests and therefore induce the routing of vehicles to optimize truck load. It is also important to note that all these operations are carried out almost continuously because orders are not known in advance and they are processed several times a day.

4 Solutions found and State of The Art

In a first approach, the problem that concerns us could be proposed as a VRP problem (Vehicle Routing Problem),

a case highly known and well-studied in the scientific literature. It is a generalization of the TSP and belongs to the combinatorial optimization problems group. The objective of the VRP is to minimize the total travel length under several constraints: the capacity constraint, the maximum number of routes constraint, travel constraints, service constraints, the time windows constraints and the maximum travel time constraint.

VRP is one of the most complicated problems from the point of view of computational complexity because it is NP-Complete. Several variations to the original problem should be sought in our specific case, because the set of pursued objectives doesn't follow a typical operational scheme already completely developed. Even so, a research in the scientific literature about MDVRP (Multi Depot VRP) and VRPTW (VRP with Time Windows) problems has been carried out. Both problems are a generalization of the VRP.

Since this is a preliminary study and the method to use is still unknown, various kinds of problems have been analyzed as well as different algorithm types for their resolution. Therefore, it will be likely to find different alternatives to solve the problem. The possibility of dividing the problem resolution into two stages has also been considered. One of them would cope with the truck loading problem, searching for the best load and truck combination. A second stage would deal with the truck routing. Both phases would pursue their own goals and would be subject to meet the global objectives.

The algorithms found have been classified into two groups: *Exact Methods* and *Approximate Methods*.

4.1 Exact Methods

The branch and bound algorithms highlight among the exact methods. Lower bounds can be set on cities combination if the problem is formulated as a linear problem, allowing this way the application of a *Branch and Bound* [1] algorithm. This type of algorithm performs solutions subsets the contribution of which to the objective function is evaluated, defining lower and upper bounds in order to decide whether or not to ramify the solution tree. Solutions are quickly obtained using these relaxations.

Padberg and Rinaldi [2] proposed an improved *Branch and Bound* (B&B), combining the classical method with a cutting plane technique and originating the *Branch and Cut* (B&C). The B&C algorithms propose the addition of new cuttings or valid inequalities as solutions to the problem.

The same authors [3] proposed the use of the above algorithm for solving the STSP problem (Symmetric TSP). In this case, it is used a polyhedral plane based on the strategy of enabling the search tree nodes share the set of restrictions, with the improvement that this fact leads from a computational point of view.

Gutiérrez-Jarpa et al. [4], proposed a variant of the B&B combining it with a column generation algorithm. This combination is known as *Branch and Price* and the authors proposed it to solve a VRPDSPTW problem (VRP, Deliveries, Selective Pickups and Time Windows). According to the results obtained, the algorithm is able to solve five variants of the problem containing up to 50 customers.

Despite the fact that exact algorithms tend to be discarded for the resolution of problems involving the handling of a large amount of data, authors like Kontoravdis [27] have proposed using them to evaluate the quality of the heuristics under study. They have used dynamic programming and B&B techniques to prove the good performance of the lower bounding methods at solving problems with the same temporal window and capacity restrictions. However, it's necessary to make use of another kind of techniques when working with more complex problems.

4.2 Approximate methods

As we mentioned before, VRP problems are computationally complex (NP-hard). This means that solving them with exact algorithms could demand a lot of time and resources. Large problems can be solved using *approximate methods* even though they will probably not provide the optimal solution but a very close one.

4.2.1 Local Search Algorithms

Local search algorithms are methods of iterative improvement which define an initial solution and a neighborhood structure. Each neighbor contribution to the objective function is explored in an iterative way in order to maximize or minimize it as appropriate.

Tabu search algorithm is a meta-heuristic that uses short-term memory, a FIFO list that stores the last visited solutions (tabu movements). These solutions cannot be visited again in the next iterations, thus avoiding cycles and eluding local minima.

Tabu search was presented by Glover and Laguna [5]. The proposal of tabu search is to look for an initial solution and efforts to exploit neighborhoods of the solution and improve optimization.

In search process, some customers are in the tabu status, e.g., a customer that leaves a route cannot change his status during a given number of iterations. The length of the tabu interval can influence the search quality. Since the initial solution of the tabu search creates from ACO, the solution is better than an initial solution randomly generated. Hence, the large Tabu interval, the half of the number of customers, is chosen to attentively explore the solution space around the initial solution in this study. Sometimes, the tabu status of a change can be overruled if the change can produce a better solution than the current best solution, or if all candidate solutions are part of the tabu list.

One of the improvements developed for the tabu search algorithm consists on adding split deliveries and time window constraints. This means that routes should meet the

established deadlines and that deliveries can be split so that a client could be served by more than one vehicle [6]. So that, each iteration of tabu search assesses the four operators of movement on the basis of the current solution and choose the best feasible neighboring solution. It may be a solution tabu or not tabu, but the tabu state will be replaced by a criterion of aspiration (better cost found so far). This method will minimize the number of vehicles and the total distance traveled.

There is another method, called *compound heuristics* [7]. Gets the name of compound because the process is performed in two distinct phases, the first one obtained through the local search local optimum, and the second by a refinement check that local optimum is global optimal. Certain conditions described in this document may be of interest to address the problem that concerns us.

Regarding the VRP problem, Pisinger and Ropke [8] solved five different problems types: CVRP, OVRP, VRPTW, MDVRP and SDVRP. They solved them using the *ANLS (an Adaptive Large Neighborhood Search Heuristic)* framework. This heuristic is an extension of the *large neighborhood search framework* developed by Shaw [9] with an adaptive layer. This layer selects among the number of insertion and removal heuristics to intensify and diversify the search. Due to the use of this layer, problems must be transformed to a *rich pick-up and delivery problem with time window (RPDPTW).* This transformation framework allows the ALNS to solve other optimization problems.

According to the results, ALNS has demonstrated to be good at solving a wide range of VRP problems, but it should be considered that transforming your specific problem into a RPDPTW type could be a difficult task.

4.2.2 Two Stages Algorithm

There is an algorithm called *Two Stages Algorithm* which basically consists of a local search algorithm. Jin, Liu, et al. [10] proposed a two-stage (TS) algorithm with valid inequalities (TSVI) that fixes a vehicle routing problem with split delivery (SDVRP) in an optimal manner. TS provides opportunities to develop strong valid inequalities in a first phase to improve the speed of computation in general. The numerical experiments show that the TSVI significantly outperforms other exact solutions contained in the literature for the SDVRP.

Chao, Golden et al. [11] presented a two-stage heuristic which consists of an initial phase of construction of solutions and a second phase to improve the solution. In the first phase, they assigned delivery day combinations to customers by solving an integer programming problem.

Lim and Zhang [12] introduced a two-stage algorithm for the VRPTW problem. The algorithm concentrates on minimizing the number of vehicles in the first stage, and on the minimization of the total travel distance in the second stage. The algorithm minimizes the number of vehicles using a data structure called the ejection pool (EP) to keep temporarily customers without service. The algorithm for minimizing the total distance is a multi-start hill-climbing algorithm using classical and defined operators including generalized ejection chains (GEC). The removal and insertion procedures are repeated to reach a new solution with less total distance.

4.2.3 GRASP Algorithm

GRASP is the acronym for *Greedy Randomized Adaptive Search Procedure* and was proposed by Kontoravdis and Bard [13] for solving the VRPTW taking into account the objective of minimizing the number of vehicles required to visit a set of nodes. Minimizing the total distance traveled is a secondary objective. Authors divided the VRPTW problem into two sub-problems depending on which objective could be relaxed; if Time Window Constraints are relaxed, it results in a bin packing problem and if Capacity Constraints are relaxed, in a scheduling problem.

GRASP combine *greedy heuristics*, *randomization* and *local* search. The algorithm has a strong difference with other meta-heuristics such as *Tabu Search* or *Simulating Annealing* because, for being more successful, GRASP needs to produce feasible solutions in the construction phase. But, on the other hand, it generates many good alternatives and this fact is an advantage for Vehicle Routing applications where unpredictable events may occur.

To assess the quality of the solution, authors implemented three different lower bounding heuristics. Results indicated that Algorithm outperforms current techniques and solves problems much faster than exact algorithms.

4.2.4 Evolutionary Algorithms

Evolutionary algorithms (EA) mimic the process of natural selection, in which, individuals with greater capacity for adaptation are those which survive. Otherwise, less skilled individuals tend to extinct.

The most common EA are *Genetic algorithms*. Basically, they define an initial population of individuals (chromosomes) and each of them represents a solution to the problem. New populations are created in every generation through crossing operators which combine selected individuals (parents) for creating new individuals (offspring). There is also a mutation operator that makes small variations to the individuals for diversifying the search space.

In genetic algorithms, there is an improvement on crossing operators. It is called *EPMX* (Extend Partial Match eXchange) and the upgrading consist in improving individuals by making them to find an arbitrary crossing position, as well as making an exchange of genes in the crossover region [14].

Likewise, Ho, Ji, et al. [15] developed a variation of genetic algorithms called *Hybrid Genetic Algorithm (HGA)*.

They solve a Multi-depot problem (MDVRP) that mixes three hard optimization problems; Grouping customers, Routing and Scheduling. These three "problems" are solved by hybridizing three heuristics in the algorithm; *Clarke and Wright saving method*, *nearest neighbor heuristic* and *iterated swap procedure*. The first two heuristics were used to generate initial solutions, whereas the last one was used to improve the solutions (parents and offspring).

In addition, two variations of the algorithm were developed: HGA1 and HGA2. The difference is given by how the initial solutions are generated; randomly in HGA1 and incorporating *C&W saving method* and *Nearest Neighbor Heuristic* into HGA2. According to the computational assess, HGA2 outperforms HGA1 in terms of total delivery time.

Berger et al. [21] presented a hybrid genetic algorithm based on removing certain customers from their routes and then rescheduling them with well-known route-construction heuristics. Some studies [19] stressed the importance of getting a good initial population and proposed a hybrid algorithm from a GA search mechanism. Most meta-heuristic algorithms also come to premature convergence, preventing them from approaching the global optima and searching the entire feasible space. Solving the difficulty of premature convergence is an important requirement and it is so important in Multi Objective Problems.

Due to the fact that VRP problems usually are multi-objective, Garcia-Najera and Bullinaria [17] proposed a *Multi-objective evolutionary algorithm (MOEA)* for solving VRPTW. They pointed out that these kind of problems aim to achieve the lowest-cost set of routes. The key is the fact that there isn´t a unique type of cost but multiples and also they are different between them (e.g. number of vehicles, delivery time or travel distance).

Results showed that solutions are more diverse and MOEA outperformed single and bi-objective algorithms. But, when MOEA was evaluated using multi-objective performance metrics (Hyper-volume and coverage), it showed better results than the multi-objective optimizer NSGA-II.

Other authors like Yu and Davis [18] affirm that meta-heuristics based on evolutionary computation have been successful dealing with multi-objective problems in many practical situations.

Other kind of multi-objective solution was presented by Wang, Li et al. [19]. The intrinsic property of a GA makes its search mechanism very effective in solving combinatorial optimization problems. They proposed a hybrid algorithm based on GA incorporating some greedy algorithms to solve the multi-objective problem (MOP) model with discrete variables.

Blanton and Wainwright [20] were the first to apply a genetic algorithm to VRPTW. To develop it, these authors hybridized a genetic algorithm with a greedy heuristic. Under this scheme, the genetic algorithm searches for a good management of customers, while the construction of the feasible solution is driven by the greedy heuristic.

Based on the philosophy of the evolutionary algorithms, there are several methods which attempt to solve problems VRP. One of them is the *Scatter Search* [16]. It is a meta-heuristic approximation, combining sets of solutions to create new ones. Unlike genetic algorithms, it uses systematic strategies on small sets of solutions.

4.2.5 Swarm Intelligence Solutions

In recent years, *Swarm Intelligence* has experimented a big growth due to the fact that many researches have focused on this promising field, thus more and more applications for those algorithms are founded.

Its objective is to mime animal collective behavior. This collective behavior happens in self-organized systems and it tries to manage or achieve vital objectives (i.e. foraging).

There has been a strong trend towards the resolution of optimization problems through the use of ACO (Ant Colony Optimization) algorithms. ACO is an optimization technic for general purpose based on the behavior of real ant colonies. Decisions for choosing the optimal solution are based on the amount of pheromone it has. As more amount of pheromone a feasible solution has, the chances of choosing it increase. This happens as real ants do, because when they are foraging, they try to find the paths with more pheromone trail as it indicates that other ant had found following it.

In the majority of these algorithms one of the most important aspects in the study of heuristic algorithms is the balance between intensification and diversification; it is necessary to accelerate the convergence and use diversification to find better solutions.

The first ACO algorithm proposed was AS (Ant System) [23], which had been successful in solving combinatorial optimization problems, but its performance is still worse than other meta-heuristics algorithms and also decreases when the size of the problem increases.

Matos and Oliveira applied the Ant System [24] in PVRP and presented a new updating strategy of pheromone information.

MMAS (MAX-MIN Ant System) [25] and ACS (Ant Colony System) [26] have been proposed as AS improvements, however, they present the possibility of staying stagnant in a local minimum producing an inappropriate steady state because the map of pheromones is dominated by a local optimum path.

Another AS improvement proposed is EA (Evolutionary Ant Algorithm) [27], which also introduces genetic mechanisms (crossing and mutation operators), as well as a meta-heuristics called NN (Nearest Neighbor) [28] that allows to obtain a global optimal solution and improve computation time.

In addition to hybrid developments studied, a new algorithm for the vehicle routing was proposed [22]. It is a hybrid approach which consists of an Ant Colony

Optimization (ACO) and tabu search. That preserves the diversity of ACO improving the search of new solutions.

Using ACO for VRP has proved to be suitable. Solving a PVRPTW type [29] is feasible with an ant solution. In spite of the complexity of having the possibility of visiting a customer on different days, it is possible to construct multi-dimensional pheromone information on each edge, i.e. there is different pheromone information on each edge during different days.

On the other hand, and following the trend of solving complex problems by simulation of the behavior of the insects, algorithms based on colonies of bees were studied. They are called ABC (Artificial Bee Colony) [30]. This kind of meta-heuristics simulates the interactive behavior of bees in the supply of food. They carried out a series of so-called movements "Waggle Dance" that allow individuals who have managed to feed sharing information about the distance, direction and amount of nectar that have been found at a certain point.

The potential scope of these algorithms is similar to others based on Swarm *Intelligence*; problems of combinatorial optimization where classic algorithms of linear optimization are ineffective. In fact, there are numerous references to problems where they have been applied and, also, applied in combination with other intelligent systems [30]. Likewise, the use of *Bees Algorithms* in a continuous optimization scenario, has proven to be more effective in the use of other meta-heuristics as PSO or Evolutionary Algorithm [31].

4.2.6 Multi-Agent Systems

The Agent-Based technology is a very promising environment to solve problems where decisions are complex and must take into account multiple variables or data. The definition of what is an agent create controversial in the artificial intelligence community [32]. Some authors, as Wooldridge [33], define agent as a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives. Others, like Hanna and Cagan [34], Franklin and Graesser [32], define them in a very similar way. In any case, it seems that there is a consensus that autonomy is a central feature and that the complexity of agent architecture lies in the interaction with the environment.

An area where Agent Technology could be implemented is Transport Logistics [35]. According to Parunak´s characteristics for an ideal application of agent technology [36], applying it to Transportation or Traffic problems with a decentralized and complex structure would aim to promising solutions.

Davidsson et al [35] also emphasized the coordination between agents. They made a classification for the coordination techniques; Control, Structure and Attitude. It is a very important feature because it established the framework for the relationships with environment. For example, the control determines if it is centralized or decentralized, while the attitude classifies agents as cooperatives or competitives.

The same authors studied references on road transports and point out that most of them concern transport scheduling, i.e., allocating transport tasks to vehicles. In this kind of problems, agents represent different roles, e.g., a company, a truck, a customer, etc. In this field, agents are usually used as a Decision Support System, which provides aid to transport operators' tactical tasks.

The aforementioned strategy of using multiple agents working together could adopt varied architectures. It depends on the relationship with environment (especially with other agents). Baker [37] studied various multi-agent architectures applied in factory control algorithms. It could be interesting for transport problems because, one of the algorithms studied is the scheduling one and it could be applied to some kind of transport problems. The main topic of Baker´s paper is the heterarchical architectures, due to the fact that it maps best onto the Internet structure because it´s heterarchical too. Also because, according to author, if we have agents widely disperse across a network and we can make this agents work together to solve a problem, they will bring together this widely distributed computing resources to come to a solution.

Hanna and Cagan [34] presented an interesting variation for Multi-agent architectures called EMAS (*Evolutionary Multi-Agent Systems*). They idea is based on the advantage that Evolutionary Algorithms have to evolve in order to achieve the best solution. Each agent is autonomous and could take decisions on its own, so each agent could work off solutions created by other agents. The choice of the best solution is distributed among the agents as they cooperate, it means that the decision making criteria could change over time to adapt to new conditions as the agent population evolves. Seldom do individuals cooperate with others in the population [37], cooperation is often considered between populations.

Authors chose TSP problem as an application of the framework presented, they explained that TSP problem allowed them to demonstrate that the framework could increase effectiveness of individual solution strategies solving a very well-known optimization problem.

According to results, the algorithm generates better solutions than individual strategies of agents. The strength of the algorithm is the fact that it can evolve the best team of agents dynamically.

5 Conclusions

A State of the Art was presented. The main objective was the study of the feasible algorithms to apply to a optimitazion problem related to a Logistics Company's fleet. Solutions, techniques and algorithms studied were mainly related to VRP or transportation problems. They were classified under two main categories; *Exact* and *Approximate Methods.*

Regarding *Exact Algorithms*, the three most important algorithms were studied. Likewise, there was presented a method for using these algorithms in the assessment of others. Exact algorithms demonstrate to be good at solving less complex problems and those with few objectives and restrictions. In other case, high computational time and resources make them not feasible.

Heuristic algorithms seemed to be a better solution when the size of the problem increases and the objectives of the optimization are varied. They usually don´t give the most optimal solution but they are very closer to, also with reasonable computing times. Generally speaking, these algorithms try to solve the problem by searching parameter combination within the solution space. The parameter combination must solve the objective function. In this way work algorithms such as *Local Search* or *Two Stages.*

Evolutionary Algorithms were developed basing on *The Natural Selection*, that means on the base of how species mutate to adapt and survive. The most known algorithms are the *Genetic Algorithms* were new populations were created based on crossing and mutating best individuals. They demonstrated to be good at solving combinatorial optimization problems and some authors proposed to use them with other heuristics to outperform the use of a single algorithm. This is the philosophy of the *Hybrid Algorithms (HA).*

In this way, *GRASP algorithm,* try to take advantage of the combination of various heuristics to solve combinational problems where different objectives must be satisfied.

The *Swarm Algorithms* is a very promising field. In our case, we study natural systems as *Ant Colonies* and *Bee Colonies.* These kinds of algorithms are used in combination with other well-known heuristics and they proved to be better than other algorithms at solving some combinatorial optimization problems.

Finally, we present another promising technology; *Multi-agent Systems.* The philosophy is very similar to *Swarm Intelligence*; the key is to have individuals that take decisions and to make them cooperate to exploit the advantages of their individual work to achieve the best solution. As before, different solutions and architectures were studied.

6 Acknowledgments

This work is supported by the Spanish Ministry of Public Works under grant MFOM-08-E12/08.

7 References

[1] Toth, P.; Vigo, D. (2000): "Models, relaxations and exact approaches for the capacitated vehicle routing problem". Discrete Applied Mathematics, vol. 123 (1-3), pp.487-512.

[2] Padberg, M.; Rinaldi, G. (1989): "A branch-and-cut approach to a traveling salesman problem with side constraints". Management Science, vol. 35 (11), pp.1391-1412.

[3] Padberg, M.; Rinaldi, G. (1991): "A Branch-and-Cut algorithm for the resolution of large-scale symmetric travelling salesman problems". Society for Industrial and Applied Mathematics REVIEW, vol.33 (1), pp.60-100.

[4] Gutierrez-Jarpa, G.; Desaulniers, G.; Laporte, G.; Marianov, V. (2010): "A branch-and-price algorithm for the Vehicle Routing Problem with Deliveries, Selective Pickups and Time Windows". European Journal Of Operational Research, vol. 206 (2), pp.341-349.

[5] Glover, F.; Laguna, M. (1997): *Tabu Search*. Kluwer Academic Publishers. Boston. USA.

[6] Ho, S.C.; Haugland, D. (2004): "A tabu search heuristic for the vehicle routing problem with time windows and split deliveries". Computers & Operations Research, vol. 31 (12), pp.1947-1964.

[7] Salhi, S.; Sari, M. (1997): "A multi-level composite heuristic for the multi-depot vehicle fleet mix problem". European Journal of Operational Research, vol. 103 (1), pp.95-112.

[8] Pisinger D.; Ropke S. (2005): "A general heuristic for vehicle routing problems". Computers & Operations Research, vol. 34 (2007), pp.2403-2435.

[9] Shaw P. (1998): "Using constraint programming and local search methods to solve vehicle routing problems". CP-98, Fourth International conference on principles and practice of constraint programming. Lecture notes in Computer Science, vol. 1520 (1998), pp.417-431.

[10] Jin, M.Z.; Liu, K.; Bowden, R.O. (2007): "A two-stage algorithm with valid inequalities for the split delivery vehicle routing problem". International Journal Of Production Economics, vol. 105 (1), pp.228-242.

[11] Chao, I.M.; Golden, B.L.; Wasil, E. (1995): "An improved heuristic for the period vehicle routing problem". Networks, vol.26 (1), pp.25–44.

[12] Lim and Zhang (2007): "A Two-Stage Heuristic with Ejection Pools and Generalized Ejection Chains for the VRPTW". INFORMS Journal on Computing, vol. 19 (3), pp.443–457.

[13] Kontoravdis, G.; Bard, J.F. (1995): "A GRASP for the Vehicle Routing Problem with Time Windows". ORSA Journal on Computing, vol.7 (1), pp.10-23.

[14] Tao, Z. (2008): "TSP Problem solution based on improved Genetic Algorithm". Proceedings of Fourth International Conference on Neural Computation, vol.1, pp.686-690.

[15] Ho, W.; Ho, G.T.S.; Ji, P.; Lau H.C.W. (2008): "A hybrid genetic algorithm for the multi-depot vehicle routing problem". Engineering Applications of Artificial Intelligence, vol. 21 (4), pp.548-557.

[16] Berger, J.; Salois, M.; Begin, R. (1998): "A hybrid genetic algorithm for the vehicle routing problem with time windows". Lecture Notes in Artificial Intelligence, Vancouver, Canada, vol.1418, pp.114-127.

[17] Wang CH.; Li CH.; Hsu Y. (2009): "Optimization of an established multi-objective delivering problem by an improved hybrid algorithm". CIE: 2009 International Conference on Computers and Industrial Engineering, vols.1-3, pp.572-577.

[18] García-Najera, A.; Bullinaria, J.A. (2011): "An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows". Computers & Operations Research, vol. 38 (1), pp.287-300.

[19] Yu T.; Davis, L. (2008): "An introduction to evolutionary computation in practice". *Evolutionary Computation in Practice, studies in computational intelligence*. Springer. Berlin. Germany.

[20] Blanton, JL.; Wainwright, RL. (1993): "Multiple vehicle routing with time and capacity constraints using genetic algorithms". Proceedings of the 5th International Conference on Genetic Algorithms. San Francisco: Morgan Kaufmann, pp. 452–459.

[21] Russell, R.A.; Chiang, W-C. (2004): "Scatter search for the vehicle routing problem with time windows". European Journal Of Operational Research, vol. 169 (2), pp.606-622.

[22] Dorigo, M.; Maniezzo, V.; Colorni, A. (1996): "Ant system: Optimization by a colony of cooperating agents". IEEE Transactions on Systems, Man, and Cybernetics, Part B, vol.26 (1), pp. 29–41.

[23] Matos, A.C.; Oliveira, R.C. (2004): "An experimental study of the ant colony system for the period vehicle routing problem". Lecture Notes in Computer Science, vol.3172, pp.286–293.

[24] Stützle, T.; Hoos, H. H. (2000): "MAX–MIN ant system. Future Generation Computer Systems", vol.16 (8), pp.889–914.

[25] Dorigo, M.; Gambardella, L. M. (1997): "Ant colony system: A cooperative learning approach to the traveling salesman problem". IEEE Transactions on Evolutionary Computation, vol.1 (1), pp.53–66.

[26] Tsai CF.; Tsai, CW. (2002): "A New Approach for Solving Large Traveling Salesman Problem Using Evolutionary Ant Rules". IEEE International Joint Conference on Neural Networks (IJCNN), vols.1-3, pp.1540-1545.

[27] Rosenkrantz, D.J.; Stearns, R.E.; Lewis, P.M. (1977): "An analysis of several heuristics for the traveling salesman problem". SIAM Journal on Computing, vol. 6 (3), pp.563-581.

[28] Yu, B.; Yang, ZZ.; Yao, BZ. (2011): "A hybrid algorithm for vehicle routing problem with time windows". Expert Systems with Applications, vol. 38(1), pp.435-441.

[29] Yu, B.; Yang, Z.Z. (2011): "An ant colony optimization model: The period vehicle routing problem with time windows". Transportation Research Part E-Logistics and Review, vol.47 (2), pp.166-181.

[30] Baykasoglu A.; Ozbakir, L.; Tapkan, P. (2007): "Artificial Bee Colony Algorithm and Its Application to generalized Assignment Problem". *Swarm Intelligence: Focus on Ant and particle swarm optimization*. Itech Education and Publishing. Vienna. Austria.

[31] Pham, D.T.; Castellani, M. (2009): "The Bees Algorithm: modelling foraging behaviour to solve continuous optimization problems". Proc. Institution of Mechanical Engineers Part C-Journal of mechanical Engineer Science, vol. 223 (12), pp. 2919-2938.

[32] Franklin, S.; Graesser, A. (1997): "Is it an agent, or just a program?: A taxonomy for autonomous agents". Lecture Notes in Computer Science, vol.1193, pp.21-35.

[33] Wooldridge, M. (2009): *Multiagent Systems*. John Wiley & Sons. West Sussex (UK).

[34] Hanna, L.; Cagan, J. (2009): "Evolutionary Multi-Agent Systems: An adaptive and Dynamic Approach to Optimization". Journal of Mechanical Design, vol.131(1), Article Number: 011010, pp.1-8.

[35] Davidsson, P.; Henesey, L.; Ramstedt, L.; Törnquist, J.; Wernstedt, F. (2005): "An analysis of agent-based approaches to transport logistics". Transportation Research part C-Emerging Technologies, vol.13(4), pp. 255-271.

[36] Parunak, H.V.D. (1999): "Industrial and practical applications of DAI". *Multiagent Systems.* MIT Press. Massachusetts (USA).

[37] Cristea, P.; Arsene, A.; Nitulescu, B. (2000): "Evolutionary Intelligent Agents". Evolutionary Computation, 2000. IEEE, vol.2, pp.1320-1328.

Comparative Analysis of Artificial Intelligence Techniques for Goods Classification

I. Fernandez[1], **D. Gonzalez**[1], **A. Gomez**[1], **P. Priore**[1], **J. Puente**[1], **J. Parreno**[1]
[1]Business Administration Department, University Of Oviedo, Gijon, Asturias, Spain

Abstract - *In this paper, different methods of inventory classification are compared. ABC classical methodology discriminates the articles to be classified according to two variables: unitary cost and yearly demand. This paper proposes different methodologies that broaden the analysis over more attributes: Genetic Algorithms, Neural Networks, Tabu Search and several techniques included the WEKA program developed by the University of Waikato. To check the reliability of the models, the results are compared to the heuristic classification that an expert made in a set of 189 pharmaceutical products considering five input attributes. In addition, an Inventory Generator Program has been used to create five inventories that have been classified by the different algorithms, so that the results obtained by the algorithms could be compared.*

Keywords: Metaheuristic, genetic algorithm, Tabu Search, Neural Network, WEKA.

1 Introduction

This paper will analyse different Artificial Intelligence techniques to solve a problem of classification of commodities. While ABC methodology discriminates the items to classify according to the variables unitary cost and yearly demand, Artificial Intelligence techniques are able to widen the analysis to more attributes.

The classification algorithms analysed in the paper are: Genetic Algorithm, Tabu search, Neural Network, Decision Tree, Multilayer Perceptron, LMT, Simple Logistic and NNGE.

The Decision Tree, Multilayer Perceptron, LMT, Simple Logistics and NNGE are all methods that are part of the WEKA Framework. The Genetic Algorithm, the Tabu Search and the Neural Network have been implemented according to several scientific papers.

The paper is organised as follows: to start with, the algorithms are described. Next, the paper explains the experimentation method that has been followed and compares the results that have been obtained. Finally, the conclusions obtained in the paper are shown.

2 ABC classification

The necessity of performing a scheduling of materials delivery to prevent inactivity situations that harm the production makes it necessary to discriminate the products to determine the ones that need a more thorough control. This is achieved by using the "ABC" classification of the past annual values of each product. It is possible to deal with the inaccuracy of the data by using a sensitivity analysis with respect to the mistakes that may appear in making product cost assignation (Chase, 1994) [1].

This classification divides the stock into 3 categories: A (very important), B (quite important) and C (little important). A product belongs to one group depending on its annual value. An item's annual value is obtained by multiplying its unitary cost (value) by its annual rotation (Cohen & Ernest, 1988) [2].

When decreasingly ordering the items' annual values and at the same time representing their accumulated annual values, a graphic is obtained that shows that a 20% of the items make up to 80% of the annual total value of the warehouse (A Class), next 50% of the items represent 15% of the annual value (B Class) and the last 30% of the items represent only 5% of the annual value (C Class). Although this classification is not always exact, it represents with enough accuracy the behaviour of items in most warehouses (Swamidass, 2000) [3].

However, there are many products that may need other relevant criteria to manage the stocks that may have more weight than the variables involved in the calculation of the product's annual value, for instance deadlines or criticality. This problem has been tackled from different approaches, among which it is worth mentioning the "Multi-attribute approach" (Flores & Whybark, 1986) [4].

Nowadays, advanced methodologies are available, that are based upon decision making. These methodologies are more accurate and less time-consuming than humans in the decision making procedure. That is why, it seems adequate the use of the Artificial Intelligence in the multi-criteria inventory classification (Guvenir & Erel, 1998) [5].

The problem of the multi-criteria inventory classification has been analysed by many researchers that propose a great diversity of methods to have this problem solved. It is worth mentioning a model of weighed linear optimisation described in "ABC inventory classification with multiple-criteria using weighted linear optimization" (Ramanathan, 2006) [18]. It is a simple model that inventory managers can easily understand. It is based on linear programming models used in data envelopment analysis. Other researchers start from a linear optimisation model that undergoes different transformations to be converted into a simpler model so that the inventory manager does not need a linear optimizer (Wan Lung Ng, 2007) [19].

Peng Zhou critics Ramanathan´s model because it might result in a situation where an item with a high value in one of the criteria is classified within class A. Peng Zhou supports a new model that uses two sets of weights for each element: one set with the most favourable weights and other with the least favourable ones (Peng Zhou, 2007) [20].

Hadi-Vencheh proposes a non-linear programming model that broadens the model described by Wan Lung Ng. This model determines a common set of weights for all elements in the inventory and keeps the effects of the weights in the final solution (Hadi-Vencheh, 2010) [21].

The attributes of the elements of the inventory are not always quantitative ones. Sometimes, linguistic expressions are used. The expressions used to evaluate an item regarding a criterion are such as Very High, High, Medium, Low and Very Low. Hadi-Vencheh proposes to integrate a fuzzy analytic hierarchic process with data envelopment analysis (FAHP-DEA). This methodology makes use of FAHP to determine the weights of the criteria and of DEA to calculate the value of the linguistic expressions (Hadi-Vencheh, 2011) [22].

The problem of ABC classification has also been tackled by using artificial neural networks (Partovi, 2002) [23]. Partovi reaches the conclusion that neural networks models have higher prediction capabilities than the Multiple Discriminant Analysis techniques.

2.1 Tabu search

Tabu search is a (TS) meta-heuristic procedure which distinctive feature is the use of adaptive memory and special strategies to solve problems. Its philosophy is based upon the exploitation of different intelligent strategies based on learning procedures to solve problems. TS's adaptive memory framework exploits the story of the problem solution process referring to four main dimensions: frequency, quality, influence and novelty (Glover, 2003) [6].

Tabu Period

Tabu Search increases the efficiency of the method of local search by using structures of memory: once a possible solution has been determined, it is marked as "tabu" so that the algorithm never visits that solution again. The number of iterations during which a solution is not considered is called "Tabu Period".

Solution Encoding

The implemented tabu search uses a vector of weights to encode the solution of the inventory classification problem. The features of this data structure are identical to those of the chromosome used by the genetic algorithm. In the same way, the fitness function used to calculate the quality of a solution is the same one that was used with the genetic algorithm.

Tabu Board and Algorithm Description

Tabu Search uses a square array with side length equal to "NumberOfCriteria+2" called "Tabu Board". The "NumberOfCriteria" is the number of features that describe each item in the inventory. Initially all cells' value in the tabu board is 0. If the value of the cell is greater than 0, it means that the cell cannot be used during the current iteration. When a cell is used, it is filled with the value of the "tabu Period". During each iteration the value of the cells is decreased by one, therefore the cells currently forbidden will be free again in the future. Cells with value 0 are not decreased and the cells in the main diagonal are not used.

The algorithm stores a "Current Solution" and a "Best Global Solution". Initially a random solution is generated. After every iteration the "Current Solution" is changed and in case this solution is better than the "Best Global Solution", the best solution is changed. In each iteration a random cell from the tabu board is selected. If this cell belongs to row I and column J, it means that the I component of the current solution is to transfer a percentage of its value to the J component. This percentage is called "Transfer Percentage". When one of the components is X_{AB} o X_{BC}, the "Current Solution" must be normalised so that the sum of the weights is 1 and $X_{BC} < X_{AB}$. Then, the fitness of the new "Current Solution" is worked out. In case the fitness is better than the one of the "Best Global Solution", the latter is updated. Finally, at the end of the iteration, the selected cell is filled with the value of the "Tabu Period" and the cells that contain positive values are decreased.

After the last iteration the algorithm returns the "Best Global Solution".

2.2 Neural Network

A neural network is formed by elements called neurones. These neurones receive series of inputs through interconnections to produce an output.

Artificial Neural Network (ANN) starts from a significant enough input data set. The goal is to have the network automatically learning the desired features. In a trained ANN the set of weights determines the knowledge of the ANN and has the property of solving the problem that the ANN has been trained to solve.

The neural network that has been implemented is the one created by Mike Gold which goal was to classify the prime numbers (Mike Gold, 2004) [7]..

2.3 Weka

The WEKA application is software that implements a collection of knowledge-machines algorithms developed at the University of Waikato (Garner, 1995) [9]. These algorithms can be used on data by using the interface offered by the software. At the same time, it can also be embedded in any application. The software includes all necessary tools to perform transformations on the data, classification tasks, regression, clustering, association and visualisation.

It is developed in java, therefore it is a multi-platform application. It is distributed under GNU-GPL license [10]. It contains different types of classification algorithms as bayes, lazy, rules and trees.

This software has been used in many researches, for instance a study on the methods of breast cancer treatment (Cakir, 2009) [13].

The algorithms of the WEKA software that have been used in this paper are: Decision Tree, Multilayer Perceptron, LMT, Simple Logistic and NNGE.

2.3.1 Decision tree C4.5

A decision tree represents the decisions made when classifying data. It is a simple tree structure, where non-

terminal nodes represent verifications of one or more attributes and the leaf node represents the final decision.

C4.5 algorithm was developed by JR Quinlan by improving an existing algorithm called ID3. The tree is generated by means of recursively-made partitions through the Depth-First strategy. The algorithm takes into account all possible tests that the set of training data can be divided into and chooses the test that produces greater information gain (Quinlan, 1992) [8].

2.3.2 Multilayer Perceptron

Multilayer Perceptron is a neural network included within the WEKA Framework. It is formed by multiple layers which help solve problems that are not linearly separable. This is a great limitation of simple perceptron.

Layers can be classified into three types:

- Input layer: it is made by those neurones that introduce input patterns into the network. In these neurones no processing takes place.
- Hidden layers: They are made by those neurones which input come from previous layers and which output go to neurones in posterior layers.
- Output layer: Neurones which output values match the whole network outputs.

The neurones in the hidden layer use the sum of the inputs as propagation rule. This sum is weighted with the synaptic weighs. A sigmoid-type transfer function is applied over the weighted sum.

The learning algorithm is called backpropagation.

This algorithm has also been used to try to infer the new interactions among members of a social network regarding the features of the users (AL HASAN, 2005) [15].

2.3.3 LMT

The "Logistic Model Trees" algorithm (LMT) consists in a decision tree structure with logistic regression functions in the leafs (Landwehr, 2005) [12].

As in an ordinary decision tree, a test on one of the attributes is associated with each internal node. For a nominal attribute with k values, the node has k secondary nodes and the cases are classified downwards in one out of the k branches depending on the value of the attribute. In the case of numerical attributes, the node has two secondary nodes and the test consists in comparing the value of the attribute with a threshold: one case is ordered downwards on the left branch if the value is lower than the threshold and on the right branch on the contrary.

Formally, a logistic model of tree consists in a tree structure comprised of a set of internal or non-terminal nodes N and a set of leafs or terminal nodes T..

2.3.4 Simple logistic

Logistic regression is a statistical method used in probabilistic classification. Simple logistic is a classifier that builds logistic regression linear models. LogitBoost with simple regression functions is used as learning base to adapt logistic models. The optimal number of iterations of LogitBoost is calculated through cross-validation which leads to the automatic attributes selection.

This method has been successfully used in Alzheimer prediction by means of biomarkers identification (Moscato, 2008) [16].

2.3.5 NNGE

Rule induction algorithms generalise the set of training samples in the form of rules. These rules can be directly evaluated to classify new instances. There are different ways of representing the rules, for instance decision trees or modular rules. Rule induction systems evaluate the attributes of the training set and decide which ones are to be used to discriminate among the different classes. Several systems have been designed with the goal of inducing rules of classification that are based on the comprehension of the structure of the data under study (Zurada, 2004) [14].

NNGE is the acronym for "Nearest-neighbor-like algorithm using non-nested generalized exemplars". It is a hybrid method between algorithms based on instances and algorithms based on rule induction. This method learns incrementally. In the beginning, it learns by classifying the instances, later on, it learns generalising each new example (Martin B., 1995) [11].

The generalisation consists in fusing the new instance together with the closest exemplar of the same class. If the closest exemplar is an isolated example, a hyper-rectangle is created that contains both exemplars. On the contrary, if the closest exemplar is a hyper-rectangle, then it increases its size so that it can include the new example.

Hyper-rectangles are represented by means of rules. In order to determine the closest neighbour, it is used an Euclidian distance function, that has been modified to be able to handle hyper-rectangles and symbolic attributes.

2.4 Main text

Experimentation has consisted in classifying Inventario189 using each algorithm. The value of the parameters of the algorithms has been varied in order to study its influence in the results. Sixty per cent of the items in the inventory have been used during the training of the algorithms. Later, five inventories have been created by using inventory generator software. The values of the parameters of the algorithms that obtained better results when classifying Inventario189 have been selected to be used to classify the five inventories with 10 iterations.

Inventario189 is the actual inventory of a pharmaceutical store. It is classified in ABC categories according to the heuristic of an expert (See Attachment 1) [25]. It contains 189 products and each product has 5 attributes:

a) Unitary Cost (€): Cost for the company to purchase the item.
b) Yearly Demand: Number of units of a given item that the company needs in a year.
c) Order Cost: Cost of replacing an item.
d) Delivery time (days): Time that passes since the order is placed until it arrives to the warehouse.
e) Criticality of the process (1-5): Level of importance of the process to which this reference belongs. The more important the process, the bigger the value. It is important in respect of the severity with which the process is affected. This may result in the

impossibility of delivering items to the client. Somehow, this attribute is related to the severity of stockouts.

Software to generate inventories has been developed. It generates a list of ABC-classified products. The generator needs the number of parameters that define each product and the upper bound (UB) and lower bound (LB) within which each parameter (P) varies depending on the class to which the product belongs. The value the parameter P can take follows a uniform probability distribution within the interval [LB UB].

The five inventories that have been generated are available in Attachment II [26]. Next, a table with their features is shown in talbe 1:

Table 1: The five inventaries

Inventories	No. of Articles Type A	No. of Articles Type B	No. of Articles Type C	Total No.of Articles
Inventory01	20	30	50	100
Inventory02	18	30	72	120
Inventory03	35	49	56	140
Inventory04	32	56	72	160
Inventory05	36	54	90	180

Next, the parameters that have been analysed in each algorithm are shown, together with the analysed values and the value that has achieved the highest success in the classification of the Inventario189. See table 2.

Table 2: results from the inventario189.

Algorithm	Parameter	Analysed Values	Best Value	Mean Success Pct.	Standard Deviation	Confidence Interval 95%	Max. Success Pct.	Min. Success Pct.
Genetic	Number of generations	50,100,150,200	100	72.854	2.28	[72.424 , 73.284]	76.72	67.725
	Population size	50,100,150	150	72.909	2.959	[72.426 , 73.392]	76.72	49.206
	Crossover Probability	0.5,0.7,0.9	0.9	73.097	2.31	[72.72 , 73.474]	76.72	67.196
	Mutation Probability	0.1,0.2,0.3	0.1	73.207	2.303	[72.831 , 73.583]	76.72	67.725
	Elitism	0,1	1	74.005	1.404	[73.818 , 74.192]	76.72	69.841
	Select two best after crossover	0,1	1	73.435	1.976	[73.171 , 73.699]	76.72	68.783
Tabu	Tabu Period	2,3,4,5	3	67.143	8.815	[63.017 , 71.269]	75.132	46.032
	Transfer Percentage	5,10,20,30	30	72.83	2.405	[71.704 , 73.956]	75.661	68.783
	Number of iterations	50,100,150,200,250	150	67.626	10.358	[62.107 , 73.145]	76.19	46.032
Neural Network	Training iterations	50,100,150,200	200	52.148	13.117	[46.734 , 57.562]	67.725	0
	Nodes on the hidden layer	2,3,4,5,6	4	53.122	12.586	[47.232 , 59.012]	67.725	6.349
	Normalisation type	1,2,3,4,5	3	59.286	2.461	[58.134 , 60.438]	66.138	55.026
Decision Tree	Confidence Factor	0.1,0.2,0.3,0.4	0.2	75.485	7.907	[73.659 , 77.311]	86.243	67.725
	MinNumObj	1,10,20	1	86.243	0	[86.243 , 86.243]	86.243	86.243
	NumFolds	1,10,20	20	75.287	8.026	[73.682 , 76.892]	86.243	67.725
	Seed	100,200	100	75.287	8.012	[73.978 , 76.596]	86.243	67.725
	Subtree Raising	0,1	1	75.287	8.012	[73.978 , 76.596]	86.243	67.725
	Unpruned	0,1	1	75.485	7.879	[74.198 , 76.772]	86.243	67.725
Multilayer	HiddenLayers	0,1,2,3,4	2	72.457	2.865	[71.521 ,	77.249	67.196

						73.393]		
Perceptron	LearningRate	0.1,0.3,0.5	0.5	72.575	2.002	[72.069 , 73.081]	77.249	69.841
	Momentum	0.1,0.2	0.1	71.323	2.005	[70.909 , 71.737]	77.249	67.196
	TrainingTime	50,100,200	200	72.734	2.014	[72.224 , 73.244]	77.249	69.841
	ValidationThreshold	20,30	20	71.299	2.018	[70.882 , 71.716]	77.249	67.196
LMT	NumBoostingIterations	-1,50,100,150,200	50	81.614	3.927	[78.331 , 84.897]	84.656	75.661
	MinNumInstances	5,15,30,40	5	82.222	5.351	[78.394 , 86.05]	84.656	67.725
	FastRegression	0,1	1	78.492	6.294	[75.546 , 81.438]	84.656	67.725
Simple	HeuristicStop	10,30,50,100	10	70.436	1.117	[70.12 , 70.752]	72.487	67.725
Logistic	MaxBoostingIterations	200,350,500	500	70.362	1.06	[70.102 , 70.622]	72.487	67.725
	NumBoostingIterations	0,50,100,150	0	70.866	1.996	[70.301 , 71.431]	72.487	67.725
	ErrorOnProbabilities	0,1	1	70.552	0.826	[70.387 , 70.717]	72.487	69.841
	UseCrossValidation	0,1	0	70.767	1.022	[70.563 , 70.971]	72.487	69.841
NNGE	NumAttemptsOfGeneOption	1,2,3,4,5,6,7,8,9,10	2	81.058	22.539	[64.935 , 97.181]	89.418	16.931
	NumFoldersMIOption	1,2,3,4,5,6,7,8,9,10	2	88.519	1.171	[87.681 , 89.357]	88.889	85.185

Table 3: Results

Inventory	Genetic Algorithm		Tabu Search		Neural Network		Decision Tree	
	Best	Average	Best	Average	Best	Average	Best	Average
Inventory01	63	61.8	48	31.8	95	93.2	97	97
Inventory02	75.833	73.5	70	59.834	71.667	69.917	92.5	92.5
Inventory03	35	35	35	32.714	77.857	74.143	92.143	92.143
Inventory04	50.625	50.188	53.125	41.375	67.5	63.562	89.375	89.375
Inventory05	77.222	76.556	76.111	56.889	85	81.278	92.778	92.778

Table 4: Statistics analysis

ANALYSIS OF THE VARIANCE OF THE RESULTS OF THE ALGORITHMS AT CLASSIFYING THE 5 INVENTORIES						
Variation Origin	Sum of squares	Degrees of freedom	Mean squares	F	Probability	F critic value
Between groups	11540.31	7	1648.616	17.51276	2.64E-09	2.312741
Within group	3012.416	32	94.13798			
Total	14552.73	39				

Table 5: Statistics analysis

ANALYSIS OF THE VARIANCE OF THE RESULTS OF THE ALGORITHMS OF THE WEKA FRAMEWORK						
Variation Origin	Sum of squares	Degrees of freedom	Mean squares	F	Probability	F critic value
Between groups	108.5473	4	27.13682	0.880002	0.493486	2.866081
Within group	616.7444	20	30.83722			
Total	725.2916	24				

The five generated inventories have been classified using the best values of each parameter in each algorithm. Every technique has been applied 10 times to each inventory. Sixty per cent of the products have been used during the training. Next, the mean and best results that have been achieved using each algorithm are shown. The table 3 sow the results of the experiments.

It can be observed that the algorithms of the WEKA Framework have achieved better percentage of success then the rest of the algorithms. The Decision Tree has been the best at classifying Inventories 2 and 3, while NNGE has been the best at classifying Inventories 1, 4 and 5. Tabu Search has achieved a low rate of success in classifying the five inventories. The Genetic Algorithm has achieved an acceptable rate of success except for the classification of Inventory 3. Finally, the Neural Network has achieved good percentages of success, especially at classifying Inventory 1.

An Analysis of Variance (ANOVA) has been performed in order to compare the different percentages of success achieved by each algorithm to determine if some of them significantly differ from the rest, table 4. The value of Alfa has been set to 0.05.

The value of F (17.51276) is bigger than F critic value (2.312), therefore, it can be concluded that at least one of the algorithms has produced results which average has statistically significant differences with the rest of algorithms.

It has been decided to perform an analysis of the variance (ANOVA) taking only into account the results of the algorithms included in the WEKA Framework (Decision Tree, Multilayer perceptron, LMT, Simple Logistics and NNGE). Results of the analysis in table 5.

In the previous table, it is observed that F (0.880002) is lower than F critic values F (2.866081). Therefore, there is no evidence of a significant difference among the five algorithms of the WEKA Framework. In other words, it is reasonable to suppose that all five methods classify inventories with an equal amount of variability. Therefore, the supposition of variance homogeneity for the ANOVA procedure is justified.

3 Conclusions

In this paper several methods of inventory classification have been compared: Genetic Algorithms, Neural Networks, Tabu Search and different techniques included in the WEKA software developed by the University of Waikato. A comparative analysis with the heuristic classification performed by an expert has been performed to check the reliability of the models. It has been selected a set of 189 pharmaceuticals and five input attributes have been analysed. Finally, the results that the different algorithms have achieved when comparing five inventories have been compared. The five inventories were created by an inventory generator program.

The algorithms in the WEKA Framework have achieved better percentages than the other algorithms. The algorithms that combine good results with high speed are the Decision Tree and NNGE. The rest of WEKA algorithms (Multilayer Perceptron, LMT and Simple Logistic) achieve good results but they use much time. The analysis of the variance (ANOVA) of the results has allowed us to determine that WEKA methods classify inventories with an equal amount of variability.

The Neural Network has achieved good results at classifying inventories 1 and 5, but its success at classifying the rest of the inventories has been low. The mean time used by the Neural Network has been low.

Tabu Search has achieved acceptable percentages of success in the classification of Inventario189, but its achieved bad results at classifying other inventories.

The Genetic Algorithm has achieved low percentages of success at classifying Inventory 3, but its success has improved at classifying the rest of the inventories.

4 References

[1] CHASE, R. y AQUILINO, N. Dirección y Administración de las Operaciones. Ed Addison-Wesley Iberoamericana. 1994.

[2] COHEN, M. A., & ERNST, R. Multi-item classification and generic inventory stock control policies. Production and Inventory Management Journal, 29 (3), 6-8. 1988.

[3] SWAMIDASS, P. M. ABC analysis or ABC classification. In P. M. Swamidass, Encyclopedia of production and manufacturing management, vol. 1-2. Boston: Kluwer Academic Publishers. 2000.

[4] FLORES, B. E., & WHYBARK, D. C. Multiple criteria ABC analysis. International Journal of Operations and Production Management. 6 (3), 38-45. 1986.

[5] GUVENIR, H. A., & EREL, E. Multicriteria inventory classification using a genetic algorithm. European Journal of Operational Research. 105 (1), 29-37. 1998.

[6] FRED GLOVER, BELÉN MELIÁN "Tabu Search". Revista Iberoamericana de Inteligencia Artificial. No.19, pp. 29-48. 2003.

[7] MIKE GOLD. "Training and Simulating a Neutral DotNetwork in C#". 2004.

[8] QUINLAN, J.R. C4.5: Programs for Machine Learning, Morgan Kaufmann. 1992.

[9] S.R. GARNER; S.J. CUNNINGHAM, G. HOLMES, C.G. NEVILL-MANNING, AND I.H. WITTEN. "Applying a machine learning workbench: Experience with agricultural databases". Proc Machine Learning in Practice Workshop, Machine Learning Conference, Tahoe City, CA, USA págs. 14-21. 1995.

[10] http://www.cs.waikato.ac.nz/ml/weka/

[11] MARTIN B. "Instance-Based learning: Nearest Neighbor With Generalization". Hamilton, New Zealand: University of Waikato. 1995.

[12] LANDWEHR, N., HALL, M. AND FRANK, E. "Logistic Model Trees". Machine Learning, Volume 59, Issue 1-2: 161-205. 2005.

[13] CAKIR A., DEMIREL B. "A Software Tool for Determination of Breast Cancer Treatment Methods Using Data Mining Approach". 2009.

[14] W. DUCH, R. SETIONO, AND J. ZURADA Computational intelligence methods for understanding of data. Proceedings of the IEEE, 92(5):771–805. 2004.

[15] MOHAMMAD AL HASAN, VINEET CHAOJI, SAEED SALEM, AND MOHAMMED ZAKI "Link Prediction using Supervised Learning". Rensselaer Polytechnic Institute, Troy, New York 12180. 2005.

[16] PABLO MOSCATO, MARTÍN GÓMEZ RAVETTI "Identification of a 5-Protein Biomarker Molecular Signature for Predicting Alzheimer's Disease". Centre for Bioinformatics, Biomarker Discovery & Information-Based Medicine, The University of Newcastle, Callaghan, Austral. 2008.

[17] CHING-WU CHU, GIN-SHUH LIANG, CHIEN-TSENG LIAO. "Controlling inventory by combining ABC analysis and fuzzy classification". Computers & Industrial Engineering. Volume 55, Issue 4, Pages 841-851. 2008.

[18] RAMAKRISHNAN RAMANATHAN "ABC inventory classification with multiple-criteria using weighted linear optimization". Computers & Operations Research 33, Pages 695–700. 2006.

[19] WAN LUNG NG "A simple classifier for multiple criteria ABC analysis". European Journal of Operational Research 177, Pages 344–353. 2007.

[20] PENG ZHOU, LIWEI FAN (2007): "A note on multi-criteria ABC inventory classification using weighted linear optimization". European Journal of Operational Research 182, Pages 1488–1491

[21] A. HADI-VENCHEH (2010): "An improvement to multiple criteria ABC inventory classification". European Journal of Operational Research 201, Pages 962–965

[22] A. HADI-VENCHEH, A. MOHAMADGHASEMI (2011): "A fuzzy AHP-DEA approach for multiple criteria ABC inventory classification". Expert Systems with Applications 38, Pages 3346–3352

[23] FARIBORZ Y. PARTOVI, MURUGAN ANANDARAJAN (2002): "Classifying inventory using an artificial neural network approach". Computers & Industrial Engineering 41, Pages 389-404

[24] YE CHEN, KEVIN W. LI, D. MARC KILGOUR, KEITH W. HIPEL (2008): "A case-based distance model for multiple criteria ABC analysis". Computers & Operations Research 35, Pages 776-796

[25] http://www.slideshare.net/DiegoGonzlezMndez/inventory189

[26] http://www.slideshare.net/DiegoGonzlezMndez/05-inventories

Forecasting S&P500 Index Movement with Support Vector Machines

R. Rosillo[1], **D. de la Fuente**[2], **and J. A. L. Brugos**[3]
[1]Department of Business and Administration and IT Department, University of Oviedo, Gijón, Asturias, Spain
[2]Department of Business and Administration, University of Oviedo, Gijón, Asturias, Spain
[3]IT Department, University of Oviedo, Gijón, Asturias, Spain

Abstract - *The aim of this research is to forecast the movement of SP&500 weekly. Support Vector Machine Classifier (SVMC) is used in order to examine the data. The data covers the period between 28/11/1997 and 29/12/2010. The inputs are technical analysis indicators such as the Moving Average Convergence Divergence (MACD) and the Relative Strength Index (RSI). SVMC is used to optimize the values of the MACD and RSI in order to determine the best situations to buy or sell the index. The two SVM outputs are the movement of the index and the probability of success to each forecast move. The best result has been achieved is a hit ratio of 92,3% using the SVM Classifier. The training data covers the period between 06/03/2009 and 06/10/2010. The testing data covers the period between 07/10/2010 and 29/12/2010. In conclusion, better results have been obtained analyzing short periods of S&P500 than using large periods.*

Keywords: Artificial Intelligence, Support Vector Machines, Technical Analysis.

1 Introduction

Exchange decision making is a subject in constant innovation. In order to generate purchase and sale recommendations, the analysts have used mainly two kind of analysis: the Fundamental Analysis and the Technical Analysis. Nowadays, Artificial Intelligence is helping investors in their decisions making. A widely used Artificial Intelligence technique is Support Vector Machines (SVM).

Technical analysis is the most used by investors [17]. This study is mainly based on the Technical Analysis and more precisely on the RSI and MACD indicators of the Quantitative Analysis. RSI and MACD are the inputs for the SVM.

The rest of the paper is structures as follows. In section 2, the analysis of the state of the art relevant to SVM and technical analysis is presented. Section 3 explains the trading rule designed. Section 4 shows the empirical results of the trading rule. Section 5 provides some concluding remarks.

2 Analysis of the state of the art

2.1 Support Vector Machines

In this section, a basic theory of the Support Vector Machine Classifier model is presented. SVM originated as an implementation of Vapnik [12]. For a detailed introduction to the subject [1], [2].

SVM are a very specific way of learning algorithms characterized by the capacity control of the decision function and the use of the kernel functions [3]-[5]. The correct election of the kernel function is very important. The methods based in kernel functions suggest that instead of attaching to each element of the input domain represented by

$$\Phi : X \rightarrow F \tag{1}$$

a function

$$K : X \times X \rightarrow R \tag{2}$$

is used to calculate the similarity of each pair of objects in the input set, an example is illustrated in Figure 1 [14].

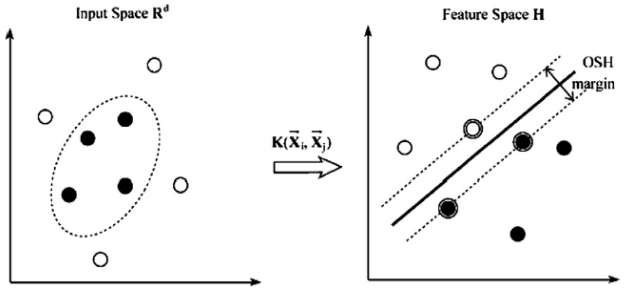


Fig. 1. An example of how a kernel function works.

Some examples of Kernel functions are:

Polynomial Function: $K(x_i, x_j) = (x_i \bullet x_j + 1)^d$ (3)

Radial Basis Function: $K(x_i, x_j) = \exp(-\gamma \| x_i - x_j \|^2)$ (4)

The biggest difference between SVM and other traditional methods of learning is that SVM does not focus on constructing hypotheses that make few errors like other techniques. SVM try to make forecasts in which the user can be very confident that the results will be correct, although it can have a lot of errors for a specific period.

Traditionally, most learning algorithms have focused on minimizing errors generated by the model. They are based on what is called the principle of Empirical Risk Minimization (ERM, Empirical Risk Minimization). The focus of SVM are different. It does not seek to reduce the empirical risk making a few mistakes, but pretend to build reliable models. This principle is called Structural Risk Minimization. The SVM search a structural model that has little risk of making mistakes with future data.

The main idea of SVM is to construct a hyperplane as the decision surface such that the margin of separation between positive and negative examples is maximized [13].

SVM can be used in different ways: regression or classification. Some applications of SVM to financial forecasting problems have been reported recently [6-11]. In this study Support Vector Machine Classifier has been utilized.

2.2 Technical analysis

Type the title approximately 2.5 centimeters (1 inch) from the top of the first page and use 20 points type-font size in bold. Center the title (horizontally) on the page. Leave approximately 1 centimeter (0.4- inches) between the title and the name and address of yourself (and of your co-authors, if any.) Type name(s) and address(s) in 11 points and center them (horizontally) on the page. Note that authors are advised not to include their email addresses.

The major literature survey on technical analysis is [16]. Technical analysis is defined how the prediction of future exchange rate (or other asset-price) movements from an inductive analysis of past movements, using either qualitative methods (e. g. recognition of certain patterns in the data for visual inspection of a time-series plot) or quantitative techniques (e.g. based on analysis of moving averages), or a combination of both [16].

Before starting to explain the evaluated articles, it is important to highlight the following statement: Technical Analysis is much more used than the Fundamental Analysis; according to [17], 90% of the polled investors use it.

The use of the Technical Analysis for the generation of trading rules is widely used nowadays. However there is something that continues to raise questioning since its inception: the knowledge of Technical Analysis can really generate capital gains. If the Trading rules in the Technical Analysis of values serve for the generation of capital gains, it implies that the hypothesis of the efficient market (which establishes that the price quoted in a period of time reflects all the available information) is not sustainable.

Studies can be grouped according to the literature in two ways of thinking: those whose position is different to the application of the Technical Analysis of values, and those that show that the use of the Technical Analysis and some indicators are able to generate greater returns than the passive strategies, for example the buy and hold strategy.

Fama [18] discovered that the study of the historical prices is not able to predict future prices and he explained the efficient market hypothesis (E.M.H.) in [19]. Neftçi [20] stated that when the economical and temporal series are assumed as Gaussians, the market indicators are not able to help to predict future prices, but when the prices are not lineal, they can show some prediction. It is also worth noting that [21] proved that the moving average used as trading systems did not always generate a greater profitability than the buy and hold strategy.

On the other hand, there are another great number of more recent studies than the previous ones which do recommend the use of the trading rules in the Technical Analysis of values like [22] who supported that when the information was not discounted on the price, the historical prices can help to achieve higher profitability. In another article [23] proved that the use of the moving averages and the use of the supports and resistances like trading tools for the Technical Analysis of values of the Dow-Jones Index from 1897-1986 generated better profitability than the buy and hold strategy in the same index. In [24] explained a similar result to the one considered in the mentioned article, but for FT30 index. Also [25] documented that the technical rules achieved a better profitability than the buy and hold strategy in the NYSE. And lastly, in [15], the oscillators RSI and MACD are evaluated in FT-30 index, and it is shown that the use of both oscillators generate a greater profitability than the buy and hold strategy. So, RSI and MACD are the inputs of the SVM in this study.

3 Trading rule designed

The Following sections explain how the design of the trading rule has been developed. An outline of it is shown in Figure 1. The software used is MATLAB 7.8.0(R2009a).

3.1.1 The data

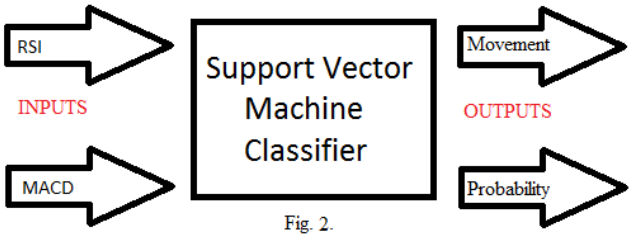


Fig. 2.

Fig. 2. Design of the trading rule.

The data base used in this study has been obtained free of charge from http://www.megabolsa.com. This data base has all the necessary information of S&P500 index.

The data covers the period between 28/11/1997 and 29/12/2010. SVM is trained with different periods of time with this data base in order to get different results and compare them. In Table I, some examples of training and testing tests are shown.

TABLE I
TEST EXAMPLES

Colour	*Training Period*	*Testing Period*	*Hit ratio (%) with C=50*
Red	*21/12/2009-11/11/2010*	*19/11/2010-29/12/2010*	*64.3%*
Green	***22/12/2008-06/10/2010***	***07/10/2010-29/12/2010***	***92.3%***
Blue	*26/12/2006-04/03/2010*	*05/03/2010-29/12/2010*	*69.6%*
Black	*29/12/1998-08/10/2009*	*09/10/2009-29/12/2010*	*58.5%*

Some examples of training and testing tests are shown in this table.

3.1.2 The inputs

The inputs of the SVM are the Relative Strength Index (RSI) and the Moving Average Convergence Divergence (MACD) indicators. These indicators have been chosen because they are the most known in quantitative analysis.

The running of the RSI and MACD indicators is summarized in [15].

-- RSI.

RSI is an indicator of the Quantitative Analysis. It is an oscillator that shows the strength or speed of the exchange rate of the price by means of the comparison of the individual upward or downward movements of the consecutive closing prices.

It was designed by J. Welles Wilder Jr. and it is explained in [25]. The main equation of the RSI is shown below:

$$RSI_t(n) = \frac{\sum_{i=0}^{n-1}(P_{t-i} - P_{t-i-1})1\{P_{t-i} > P_{t-i-1}\}}{\sum_{i=0}^{n-1} | P_{t-i} - P_{t-i-1} |} \times 100; \quad (5)$$

Where RSI is Relative Strength Index at time t, Pt is the value of index at time t, n is the number of RSI periods, 1{_} is an indicator function which equals one when the statement inside the bracket is true and equals zero otherwise, |x| is the absolute value of x. The 14-day RSI, a popular length of time utilized by traders, is used in this study. The RSI ranges from 0 to 100 however the range has been reduced from -1 to +1 in order to place it in the SVM.

-- MACD.

MACD is an indicator of the Quantitative Analysis. It is designed mainly to identify the tendency changes.

The MACD is constructed based on the moving averages. It is calculated by subtracting the longer exponential moving average (EMA) from the shorter EMA. The EMA is defined as:

$$EMA_t = \left[\frac{2}{n} \times (P_t - EMA_{t-1})\right] + EMA_{t-1}; \quad (6)$$

Where EMAt is the exponential moving average at time t, n is the number of periods for EMA. The initial EMA is the n-day simple moving average of the series. In this article, we focus our attention on the 12 and 26-day EMAs, which are the most commonly used in order to calculate MACD[26].

The range of MACD has been reduced from -1 to +1 in order to place it in the SVM.

3.1.3 The trading rule

The trading rule is managed by the SVM Classifier:

First, the SVMC analyses the inputs RSI and MACD.

Second, SVMC classifies the different prices of the S&P500 in two classes: buying and selling situation. The prediction made by the SVM is weekly.

Third, SVM also plot the conditions of purchases and sales, shows the likelihood that the prediction will be fulfilled. Thus, the very risk averse investor will choose a

high probability of success, while a little risk averse investor will choose a lower probability of success.

Fourth, the hit ratio is calculated with the probabilities selected.

Finally, SVM optimize the buying and selling values of input variables (RSI and MACD). That is, given a value of RSI and a value of MACD, SVM predict the upward or downward movement every week and the probability that this occurs.

3.1.4 The trading outputs

The outputs of SVM are the movement that made the index next week, and the probability that this movement is accomplished.

- Movement to make the index.

The movement that the outputs of the SVM indicate it will be bullish or bearish.

- The probability that this movement is accomplished.

The investor can set the SVM to do the trading on a given probability. This allows the trading rule is used by all investors, both risk averse as they are not.

4 Results

There have been various types of tests with the SVM, and the most relevant are displayed below.

In Figure 3, a comparison of results obtained in 4 tests performed is shown. The study period in Figure 3 are described in Table 1.

As can be seen, the best results are produced with a short training period. On the other hand, the worst results are produced with long training period. The vertical axis shows the hit ratio in% obtained by the SVM in the test period and the horizontal axis shows the value of the parameter C of SVM. Better results are produced with a lower value of the C parameter.

It has been taken into consideration that the investor is risk averse and only run commands that have a higher probability than 90% accuracy. The HTRBF (Heavy Tailed Radial Basis Function) kernel function has been utilized in the SVM algorithm.

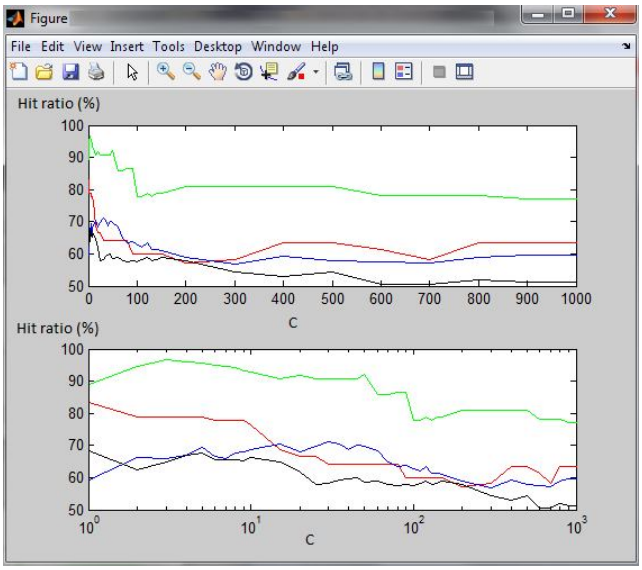


Fig. 3. Comparison between long testing period and short testing period.

Figure 4 shows the testing period for training data between 22/12/2008 and 06/10/2010. The value of C parameter is 50.C=50.

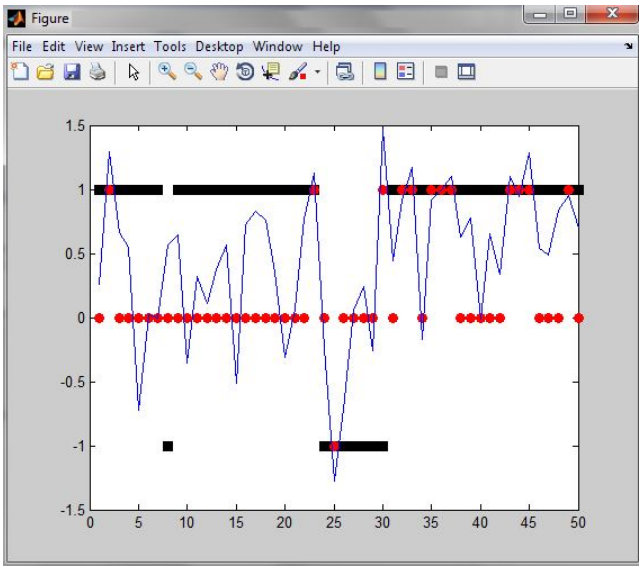


Fig. 4. Testing period for training data between 22/12/2008 and 06/10/2010.

Black squares indicate the desired response of the SVM. The blue line impinges for each day of testing the value of the probability of success of that particular day. The red dots indicate the action taken by the SVM, since for this case only takes into consideration the results with a probability of success greater than 90%, the SVM gives 13 recommendations, of which there are 12 hits and 1 error in prediction.

5 Conclusions

This study shows that SVM produces better results using shorter training periods than long training periods. This may be caused by overtraining. It has managed to achieve good results for trading, the best situation it is a 92.3% of hit ratio. Moreover, the developed software allows the user to determine their level of risk aversion, which lets the user distinguish the study for each type of investor.

6 Acknowledgment

Financial support given by Government of the Principality of Asturias is acknowledged.

7 References

[1] C. Burges. "A tutorial on support vector machines for patter recognition". Data Mining and Knowledge Discovery, 1998, vol. 2, pp. 121-167.

[2] T. Evgeniou, M. Pontil, T. Poggio. "Regularization networks and support vector machines". Advances in Computational Mathematics, 2000, vol. 13, pp.1-50.

[3] V. N. Vapnik. "Statistical learning theory". New York: Wiley, 1998.

[4] V. N. Vapnik. "An overview of statistical learning theory". IEEE Transactions of Neural Networks, 1999, vol. 10, pp. 988-999.

[5] N. Cristianini, JS. Taylor. "An introduction to support vector machines and other kernel-based learning methods". New York: Cambridge University Press, 2000.

[6] L. J. Cao, F. E. H. Tay. "Financial forecasting using support vector machines". Neural Computing Applications, 2001, vol. 10, pp. 184-192.

[7] F. E. H. Tay, L. J. Cao, "Application of support vector machines in financial time series forecasting". Omega, 2001, vol. 29, pp. 309-317.

[8] F. E. H. Tay, L. J. Cao. "A comparative study of saliency analysis and genetic algorithm for features selection in support vector machines". Intelligent Data Analysis. 2001, vol. 5, pp. 191-209.

[9] F. E. H Tay, L. J. Cao. "Improved financial time series forecasting by combining support vector machines with self-organizing feature map". Intelligent Data Analysis. 2001, vol. 5, pp. 339-354.

[10] F. E. H. Tay, LJ. Cao. "Modifies support vector machines in financial time series forecasting". Neurocomputing. 2002, vol.48, pp. 847-861.

[11] W. Huang, Y. Nakamori, S.Y. Wang. "Forecasting stock market movement direction with support vector machine". Computers & Operations Research. 2005, vol. 32, pp. 2513-2522.

[12] V.N. Vapnik. "The nature of statical learning theory". New York. Springer.

[13] X. Xu, C. Zhou, Z. Wang. "Credit scoring algorithm based on link analysis ranking with support vector machine". Expert Systems with Applications, 2009, vol. 36, pp. 2625–2632.

[14] S. Huang, Z. Sun. "Support vector machine approach for protein subcellular localization prediction". Bioinformatics. 2001, vol. 17(8), pp. 721-728.

[15] Terence Tai-Leung Chong, Wing-Kam Ng ing-Kam. "Technical analysis and the London stock exchange: testing the MACD and RSI rules using the FT30". Applied Economics Letters. 2008, vol. 15:14, pp. 1111–1114.

[16] L. Menkhoff, M.P. Taylor. "The obstinate passion of foreign exchange professionals: technical analysis". Journal of Economic Literature. 2007, vol: 45, pp. 936–972.

[17] M.P. Taylor, H. Allen. "The use of technical analysis in the foreign exchange market". Journal of International Money and Finance. 1992, vol. 11, pp. 304-314.

[18] E.F. Fama. "The behavior of stock-market prices". Journal of Business. 1965, vol. 38, pp. 34-105.

[19] E.F. Fama. "Efficient capital markets: Review of theory and empirical work". Journal of Finance. 1970, vol. 25, pp. 383-423.

[20] S.N. Neftçi. "Naïve trading rules in financial markets and Wiener-Kolmogorov prediction theory: A study of 'technical analysis". Journal of Business. 1991, vol. 64, pp. 549-571.

[21] R. Hudson, M. Dempsey, K. Keasey. "A note on the weak form efficiency of capital markets: the application of simple technical trading rules to UK stock prices — 1935 to 1994". Journal of Banking and Finance. 1996, vol. 20, pp. 1121–1132.

[22] J.L. Treynor, R. Ferguson. "In defense of technical analysis". Journal of Finance. 1985, vol. 3, pp. 757-773.

[23] W. Brock, J. Lakonishok, B. LeBaron. “Simple technical trading rules and the stochastic properties of stock returns”. Journal of Finance. 1992, vol. 47, pp. 1731–1764.

[24] T.C. Mills. “Technical analysis and the London Stock Exchange: Testing trading rules using the FT30”. International Journal of Finance and Economics. 1997, vol. 2, pp. 319–331.

[25] J. Jr., Welles. “New Concepts in Technical Trading Systems”. 1978.

[26] J.J. Murphy. “Technical Analysis of the Financial Markets”. New York Institute of Finance. 1999

Using Cloud Computing with RETE Algorithms in a Platform as a Service (PaaS) for Business Systems Development

Richard Poblete[1], **David de la Fuente**[2], **and Margarita Alonso**[3]
[1]Department of Business Administration, Cantabria University, Santander, Spain
rpoblete@postgradouchile.cl
[2]Department of Business Administration, Oviedo University, Gijón, Asturias, Spain
david@uniovi.es
[1]Department of Business Administration, Cantabria University, Santander, Spain
alonsom@unican.es

Abstract - *Given that companies operate in an environment of continuous evolution it requires information systems to have greater adaptation, flexibility, scalability and reliability of incorporation of these changes. Cloud Computing can be regarded as a paradigm and key element of recent computing in business management solutions. A business solution arises in the cloud; this solution includes both the environmental dynamics and the changes needed in the business logic assuring speed of response, efficiency in the solution and maintenance of the application. This article shows a design model of Saas applications based on a platform (PaaS) which has the business logic as central element and is subdivided in "Business Objects", "Business Processes" and "Business Rules." According to the business rules, and in order to resolve the problem based on facts and inferences from the knowledge as an essential part of the construction of the system, expert systems are applied through the use of the RETE algorithms (Forward Chaining).*

Keywords: Cloud Computing, SaaS, PaaS, RETE

1 Introduction

The competitive business environment demands companies to concentrate in business management and not in the technological aspects of their company which may be complex because of a changing environment. It is therefore necessary to constantly redefine the Business Logic Systems. Having a platform which can adapt quickly to these conditions is extremely important.

Cloud Computing is changing the traditional way of how hardware and software providers sell their products to their customers. This is another kind of business in which the product is not sold as such, but a service is sold [2]. Mainly, this platform allows sharing resources as a service. These resources include infrastructure (IaaS), software (SaaS), applications (PaaS), and data storage (dSaaS). At the heart of this technology is the Virtualization and its model of operative business is varied and can include: Pay per Use Models; Free Models; Fee for Service Models and Free Use Models, sharing utilities.

1.1 Cloud Computing

Cloud Computing is defined by the way in which the service provider and the consumer of that service are related. In this context it is very common that the consumers are not the owners of the infrastructure, they do not incur CAPEX[1], since they are under a modality of rent and use from a provider. The customers consume resources and only pay for resources used. This model is very similar to what happens with the payment and consumption of energy in our homes. This business model is possible because of the following features:

1) Virtualization: Cloud Computing allows the user to access the service in any place through any kind of terminal or device. The required resources come from the Cloud instead of a visible entity.

2) Reliability: The Cloud uses various methods for the fault tolerance. Computing based on interchangeable isomorphic nodes guarantees the service reliability. The Use of the Computing Cloud is more reliable than in a local station [2].

3) Versatility: The Cloud can stand different applications, execute them at the same time and reuse their own resources.

4) Elasticity: The Cloud scale can expand dynamically in order to satisfy the growing demand for requirements.

5) "On-Demand" Service: The Cloud is a wide group of resources that can be rented according to the customer necessity.

[1] CAPital EXpenditures (CAPEX or capex) are capital inversions or expenditures which create benefits.

6) Economic: cost reduction is given by the fault tolerance feature which has the Cloud "per se", (very cheap isomorphic nodes). The versatility, elasticity and "On Demand" Service make the resources' availability easy [2] compared to the traditional system, so the user can enjoy this low-cost advantage.

The article starts explaining the relevant concepts, a proposed method, and analyzing the implications of using RETE as an essential part of such a model.

2 Expert Systems (ES)

Expert systems as techniques within the area of Artificial Intelligence are presented as computer programs that can perform various tasks:

- Efficiently carrying out search and recovery processes for interrelated information stored on a large scale.
- Formal representation of symbolic knowledge on the field in which the system acts.
- Allowing inferences and heuristics to complete knowledge.
- Explanation of the conclusions reached, interacting with the user explaining the line of reasoning pursued.

Expert Systems constitute efficient help when requiring a good interpretation of complex information, advice on the field in which the expert system works, demanding comparable output from the system to that from human experts.

Expert Systems are based on the declarative knowledge (facts about objects, situations) and control knowledge (information about the monitoring of an action). [3] [4]. Expert Systems incorporate capacities of explaining the reasoning followed in the resolution of problems and the acquisition of new knowledge, and also they present simple interaction procedures with the users [9].

3 RETE Algorithm

The RETE algorithm is a pattern matching algorithm for implementing production rule systems. It was designed by Dr. Charles L. Forgy of Carnegie Mellon University. RETE has become the basis for many popular expert systems, including CLIPS, Jess, Drools, and Soar.

The RETE algorithm provides a logical description responsible for matching data tuples ("facts") against rules in a pattern-matching production system. A production rule system consists of one or more conditions and different actions which may be undertaken for each complete set of facts that match the condition. The RETE algorithm RETE exhibits the following major characteristics:

- It reduces or eliminates certain types of redundancy through the use of node sharing.
- It stores partial matches when performing joins between different fact types. At the same time, it allows production rule systems to avoid complete revaluation of all facts each time changes are made in the nodes which are stored in the working memory. Instead, the production system has to evaluate the changes (deltas) in the working memory.
- It allows for efficient removal of memory elements when facts are retracted or moved from working memory.

The RETE algorithm is widely used to implement matching functionality within pattern-matching engines to support the Chain Forward[2] and useful to interface in the formulation and use of the business rules.

4 PaaS Model focused on the business logic

We propose a Cloud Computing model [8] that attempts to conceptualize the development of a platform as a service (PaaS) to support software solutions as a service (SaaS). The model is essentially based on the business layer, taking into account that it is in this good level where the development of the main processes and algorithms of the SaaS solution are required, leaving a secondary and subordinated rule to the other two layers (Data and interface) in the business layer.

In the model, the interface will depend on how we access the SaaS solution (as Internet navigators, mobile devices, business logic web services). Data layer is only the object persistence, the metadata storage of the process structures and the business rules. Figure 1 presents the general scheme of the model:

[2] Forward chaining is one of the main methods of reasoning when rules of inference (in artificial intelligence) are used and can be logically described as repeated applications of "modus ponens". Its implementation in expert systems and business and productive rule systems is popular.

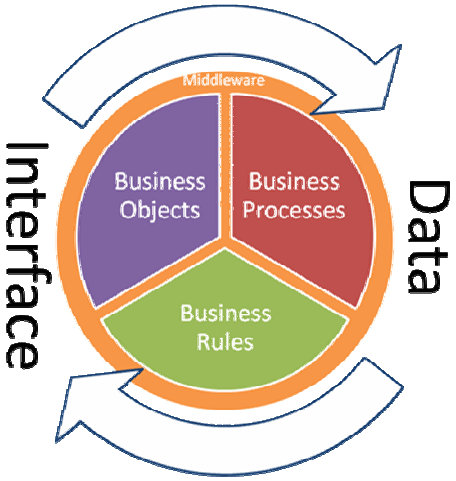


Figure 1: Model based on the business logic

4.1 Business Logic Layer:

It contains the heart of the system. It is a group of layers with subroutines developing the last goal of the application. It uses interfaces among layers and management interfaces oriented to the user in order to produce the parametrical adjust of the Business Logic Layer to the requirement changes in the environment and within the organisation.
Business Logic Layer is composed of the following interrelated sub-layers:

4.1.1 Business Rules

It contains the group of business rules and interferences corresponding to the service implementation. This layer has user tools through the interface layer and this allows for a quick modification when changes are made in the business conditions. This layer is a production rule system, that is, a Business Rule Management System-BRMS[3], which allows us to configure externally business rules to the application using them, without modifying the programming code. One example is the use of the RETE[4] algorithm.

4.1.2 Business Objects

Business Objects are representations of organizational concepts and collaborate with others in order to reach business goals [12]. Business modelling will be given by the relations of the objects, properties and methods, and also the interaction with the business process layer and business rule layer.

4.1.3 Business Processes:

The Workflow Reference Model defines a business process as a group of one or more connected activities, which collectively make a business goal, normally within the functional structure of an organisation [11]. Business Processes are responsible for organizing the resources of the system by means of the integration of the business rules and business objects and workflow, understanding workflows not only as automata tasks through forms of task assignation to users or roles. The model proposes the management of these BPM[5] processes which are implemented and modelled through BMPN[6].

4.1.4 Interface and Data Layer

Interface Layer consists of interfaces or screens showed in the browsers, mobile devices or interfaces to other systems that consume these services. This presentation layer makes reference to the web service interface and sends encapsulated data in XML messages that only the interface knows. This layer must know the different kind of data and messages that can be received by the service interface.
Data Layer is the responsible for passive storage of the object persistence, process structures and businesses rules. The Data Layer is responsible for the structure definition storage of the business logic layer and associated data.

5 Business Rule Layer: RETE Algorithm main actor

In traditional system development, a change in the business conditions or rules would involve corrections, additions or updates in code lines, database objects, and stored procedures and in functions or triggers [5]. Complexity associated both with recoding and impact on the system design and test times can confer instability to the system maintenance and exploitation [10]. Under this development scheme, system maintenance becomes more complex and tedious as the environment changes increasingly. A BRMS assures that the system adapts to the business rule dynamism and definition modifications, without recoding programs and objects.
A Cloud Computing system, in accordance with the features "Scalability", "Versatility", "Reliability" and "Elasticity", must be efficient in all elements the system is composed of. This ensures that when incorporating an expert system within the BRMS business rule layer it must be very efficient [10]. A simple implementation of an expert system would prove each rule with the facts of the knowledge base, activating the rule, if appropriate, and evaluating the following in a process that can

3 BRMS or Business Rule Management System is a software system used to define, deploy, execute, monitor and maintain the variety and complexity of decision logic that is used by the operational systems within an organisation or enterprise.
4 The RETE algorithm is a pattern matching algorithm which is efficient for implementing a rule production system. It was created by Dr. Charles L. Forgy of the Carnegie Mellon University.

5 Business Process Management or BPM is the business methodology whose goal is to improve the efficiency through the systematic management of the business processes which should be continuously modelled, automatized and optimized. As its name suggests, BPM is focused on the management of the business processes.
6 Business Process Modeling Notation or BPMN is a common graphical notation to bridge the communication gap that frequently occurs between business process design and public and private processes, orchestration, choreography, etc. as well as advanced modelling concepts.

be iterative. This algorithm, even for a low number of rules and facts, can have a very high execution time (making it inappropriate for real production systems in the Cloud). Therefore, the utility of applying the RETE algorithm to the recent system is given by its speed and efficiency [6] and it is consequently used as a basis for more efficient expert system implementations in the Cloud.
A RETE-based expert system builds a network of nodes, where each node (except the root) corresponds to a pattern occurring in the business condition. Therefore, the path from the root node to a leaf node defines a complete rule conditional part [7]. Each node has a memory of facts which satisfy that pattern. This structure is essentially a special case of Trie[7] deterministic finite automaton.[8]
As new facts are asserted or modified, they propagate along the network, causing nodes which are implied in the pattern to get active. When a fact or combination of facts causes all of the patterns for a given rule to be satisfied, a leaf node is reached and the corresponding rule is triggered.

5.1 Limitations and Indications

When developing a business solution SaaS on a PaaS platform, the considerations that we have to take into account and that take part of the limitations of a RETE-based BRMS are:

- *There is a limited relationship between conditions in an engine of this kind and the Business Logic problem.*
- *RETE regular models are based on a structure of facts which are represented by simple groups of 3 elements which can not cover complex business logic.*
- *There are efficiency problems with large volumes of data matching* [10].
- *RETE sacrifices memory in order to implement processing speed. Memory is a poor resource in the system infrastructure.*

The first two limitations are resolved in the proposed model centred in the business logic and presented in Figure 1. The model resolves limitations through the use of a BRMS, complementing with Business Objects its attributes and methods (orientation to objects), and process automation and orchestration through BPM.
According to the use of the memory and the efficiency of the algorithm with great load of data, RETE (Forward Chaining) sacrifices memory for increased speed. In most cases, the speed increase compared to the simple implementation is several orders of magnitude (because RETE performance is theoretically independent of the number of rules in the system). However, in very large expert systems, RETE usually presents problems because of its great quantity of energy consumption. When implementing systems of great importance as a service (SaaS) and related to the use of the memory, limitation can be resolved with the "reliability" feature, that is, computing based on interchangeable isomorphic nodes which let generate parallelisms [1] resolving the problem of memory limitation, Workaround solution[9].
The proposal of improved solution, according to the previous one, is the use of optimized RETE algorithms, such as "RETE Plus Algorithm" [10].

6 Conclusions

A Cloud Computing solution in SaaS requires having a steady, elastic, scalable and reliable platform for the adequate delivery of a quality service. The proposed model integrates the use of a BRMS (RETE algorithm), with Business Objects and Business Processes which constitute the basis of an efficient PaaS platform.

A RETE-based BRMS assures that the system which is defined according to the proposed model, adapts itself to the dynamism of the environment, through a definition of flexible network of nodes which models the business rules and proposes an adapted solution to the scalability and elasticity features searched in a PaaS platform.

7 References

[1] A Computing Mode Suitable for Medium and Small-sized Enterpresis Cloud Computing [Journal] / auth. Wie Liu Liyan Zhang, Chunying Wu, Hongli Cai, Lele Qin. - [s.l.] : IEEE Computer Society, 2010. - 2010 Internacional Conference on Intelligent Computation Technology and Automation. - 978-0-7695-4077-1/10.

[2] Cloud Computing Research and Development Trend [Journal] / auth. Shuai Zhang [y otros]. - [s.l.] : IEEE Cpmputer Society, 2010. - 2010 Second International Conference on Future Networks. - 978-7695-3940-9/10.

[3] Expert Systems, Principles and Programming [Book] / auth. Giarratano Joseph C. and Riley Gary. - 2005. - ISBN 0-534-38447-1.

[4] Introduction to Expert Systems [Book] / auth. Gnizio James. - 1991. - ISBN 0-07-909785-5.

[7] A Trie is a special case of deterministic finite automaton (Set of states, alphabet which defines the chains, transition function, initial state, set of acceptation state) and is used to store a set of chains.
[8] A finite automaton or finite state machine is a mathematical model which calculates automatically an input in order to get an output. This model is composed of an alphabet, a group of states and a group of transactions among such states.

9 A Workaround is a temporal solution for a known problem, while a definitive solution is searched, if this solution exists.

[5] Match Algoriths for generalized RETE networks [Journal] / auth. Lee H. S. and Schor M. L. // Artificial Intelligence. - Vol. 54. - pp. pp. 249-274.

[6] On the efficient implementation of production systems. Ph.D. Thesis, Carnegie-Mellon [Report] / auth. Forgy Charles. - 1979.

[7] RETE : A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem [Journal] / auth. Forgy Charles // Artificial Intelligence. - 1982. - 19. - pp. 17-37.

[8] Tesis de Máster Oficial de Empresas y Tecnologías de la información [Informe] = Tesis de Máster / aut. Poblete Muñoz Richard. - 2010. - págs. 14-17; 28-45.

[9] The Essence of Expert Systems. Pearson Education [Book] / auth. Darlington Keith. - 2000. - ISBN 0-13-022774-9.

[10] The Improvement Research on Rule Matching Algorithm RETE in Electronic Commerce Application Systemas [Journal] / auth. Wang Dongyun and Ren Zhijun. - Shanghai, China : IEEE, 2008. - 978-1-4244-2108-4.

[11] The Workflow Reference Modelo [Report] / auth. Workflow Management Coalition. - 1995.

[12] Using Roles and Business Objects to Model and Understand Business Processes [Journal] / auth. Artur Caetano António Rito Silva, José Tribolet // 2005 ACM Symposium on Applied Computing. - 2005. - pp. 1308-1313.

A Fuzzy Linguistic Model for Generating Similar Short Queries

Jesus Serrano-Guerrero, Francisco P. Romero, Emilio Fernandez-Viñas, Jose Ruiz-Morilla and Jose A. Olivas
[1]Department of Information Technologies and Systems , University of Castilla La Mancha, Ciudad Real, Spain

Abstract— *This work presents a model for generating a set of queries useful for obtaining massive information from Web search engines. Given an information request expressed following a boolean structure, the proposed model generates a set of related queries that are useful for being submitted to one or several search engines in order to collect a huge amount of information. This model is based on a definition of the user query with linguistic constraints that allow guiding the process for selecting the best queries for a search process. The model presents a reformulation process which interchanges query terms using a knowledge source that provides related terms with respect to (w.r.t.) the original query terms. After reformulating the original query, a filtering process is proposed based on the aggregation of the inserted semantics to the different query terms through linguistic values. The final result of the filtering is the set of appropriate queries that can be submitted to a search engine in order to retrieve a huge amount of documents.*

Keywords: Query reformulation, Hierarchical concept, Fuzzy linguistic modeling

1. Introduction

Information Retrieval studies the way of retrieving information requested by a user according to his own criteria. The concept of Information Retrieval is usually connected to the idea of searching a precise answer. The user expresses his query and the system tries to find exactly the answer for that query. Users usually prefer a unique answer with the required information rather than a set of answers where the demanded information is contained. However the user does not always want precise information, there exist situations where a huge amount of information is necessary and a precise answer is not enough for the user request. There are systems which need great amounts of information, for example thousands of documents are necessary in order to perform a text mining process.

The aim of this work is the generation of a set of queries from an initial user query which may be able to retrieve a huge number of relevant documents from the Web. This work presents a model which allows generating thousands of queries useful for retrieving documents w.r.t. an initial user query. This initial query is represented by a weighted boolean query which presents several constraints that the user has to define by fuzzy linguistic values. The semantics of each weight will determine the process for selecting the best queries. This model needs a knowledge source which provides terms related to the query terms and which allows performing a reformulation process. The query reformulation process consists in simulating the typical behavior of Internet users who change or add words to the initial query when the system does not find the required answer until the information is found. This reformulation process generates a huge number of queries that can be more or less relevant depending on the terms that have been used to reformulate each query. Thus depending on the constraints defined by the user to the original query and the properties of the reformulated terms, a bottom-up process is computed whose goal is to analyze each reformulated query in order to determine the relationship degree w.r.t. its corresponding original query. The queries that are more similar w.r.t. the original one will be the selected ones to be submitted to one or several search engines in order to retrieve millions of documents that can be used to perform processes such as text mining.

The rest of the paper is organized as follows: section 2 presents a brief introduction to the linguistic modeling and strategies to transform a query into new queries. Section 3 comments the three main steps of our model: definition, reformulation and filtering. Section 4 presents a brief example of the application of the proposed model. Finally some conclusions are pointed out.

2. Preliminaries

2.1 Query expansion

Query expansion/reformulation is one of the most studied techniques during last years achieving great results. According to Jansen [1] query reformulation is the process of altering a given query to improve the search or retrieval performance. It is considered as a technique for dealing with the issue of word mismatching in information retrieval. There are several strategies about query expansion: manual, interactive or automatic. Efthimiadis developed a lot of experiments about interactive query expansion. Several term ranking algorithms were analyzed and their performance in an interactive environment in [2]. Using the Cranfield collection, Harman [3] studied the effectiveness of the query expansion and reweighting process, while Jones studied the query substitutions [4].

2.2 Query reformulation by web users

Query expansion can have sense specially with short queries. Short queries on the Web are a well-known phenomenon [5], because the vocabulary problem, discussed in [6], is very severe when the user queries are too short. For this reason several query formulation aids have appeared in conjunction with web-based search engines. Some users used only one query in their session; others used several successive queries. The average session is 2.84 queries for each session [5].

Many studies are focused on the behavior of web users when they reformulate their queries in order to improve the results retrieved by each search engine. Bruza [7] categorized query reformulation into 11 types and found that users frequently repeated a query that they had already submitted. Rieh [8] classified the different kinds of reformulation into six categories: (i) generalized reformulation, (ii) parallel reformulation, (iii) specified reformulation, (iv) dynamic reformulation, (v) alternative reformulation and (vi) format reformulation.

2.3 Fuzzy linguistic approach

The fuzzy linguistic approach is an approximate technique appropriate for dealing with qualitative characteristics by means of the use of linguistic variables [9]. In this approach for solving a particular problem it is necessary to choice the linguistic term set and its semantics, and the appropriate aggregation operator [10]. When the linguistic term set and its semantics have been chosen, two possibilities for choosing the appropriate linguistic descriptors of the term set can be found: the classical fuzzy linguistic approach and the ordinal fuzzy linguistic approach. The first one defines the linguistic term set by means of a context free grammar and the semantics of linguistic terms is characterized by membership functions [9], whereas the ordinal approach defines the linguistic term set by means of an ordered structure of linguistic terms, and the semantics of linguistic terms is derived from their own ordered structure [11].

If the linguistic term set is defined by means of a grammar then it is necessary to define the primary terms, the modifiers and the semantic membership functions for them, the production rules and the action semantic rules for the modifiers [12]. In an ordinal characterization of the linguistic term set all terms are assumed to be primary ones, and distributed on the interval $[0, 1]$ on which a total order is defined [11]. In this case, the semantics is introduced from the structure defined on the linguistic term set.

Let $S = \{s_i\}$, with $i \in \{0, \ldots, T\}$, be a finite and totally ordered label set in the usual sense, depending on the distribution of the linguistic terms on the interval $([0, 1])$ there are two possibilities for defining the semantics of the linguistic term set: symmetrically and non-symmetrically distributed terms.

The subdomains of the terms are described by membership functions. The use of linear trapezoidal membership functions is considered good to capture the uncertainty of the linguistic assessments. This representation is achieved by a $4 - tuple$ (a, b, c, d), where a and b determine the interval in which the membership value is 1.0 and the other two parameters determine the support. Besides we need the following operators:

- $Neg(s_i) = s_j, j = T - i.$
- $MAX(s_i, s_j) = s_i,$ if $s_i \geq s_j.$
- $MIN(s_i, s_j) = s_i,$ if $s_i \leq s_j.$

Aggregation operators are necessary in order to manage linguistic information. There exists a special kind of operators which deal with weighted information. Linguistic weighted operators aggregate linguistic information according to different criteria. Different families of linguistic aggregation operators can be found [11], [13], two examples of these families are:

Definition 1: The Linguistic Weighted Disjunction (LWD) operator allows the aggregation of a set of linguistic weighted opinions, $\{(c_1, a_1), \ldots, (c_m, a_m)\}$, $c_i, a_i \in S$ and is defined as follows:

$$LWD[(c_1, a_1), \ldots, (c_m, a_m)] = MAX_{i=1,\ldots,m} LC_k^{\rightarrow}(c_i, a_i),$$

where a_i is a weighted opinion, c_i is the importance degree of a_i, and $LC_k^{\rightarrow}$ are a group of linguistic t-norms that are monotonically nondecreasing in the weights and satisfy the properties required for any transformation function h. An example of these functions is the classical MIN operator: $LC_1^{\rightarrow}(c, a) = MIN(c, a)$.

Definition 2: Linguistic Weighted Conjunction (LWC) operator is used for the aggregation of a set of linguistic weighted opinions, $\{(c_1, a_1), \ldots, (c_m, a_m)\}$, $c_i, a_i \in S$, and is defined as follows:

$$LWC[(c_1, a_1), \ldots, (c_m, a_m)] = MIN_{i=1,\ldots,m} LI_k^{\rightarrow}(c_i, a_i),$$

where $LI_k^{\rightarrow}$ are a group of linguistic implication functions which are monotonically nonincreasing in the weights and satisfy the properties required for any transformation function h. An example of these functions is the Kleene-Dienes's implication function: $LI_1^{\rightarrow}(c, a) = MAX(Neg(c), a)$.

3. Fuzzy model for generating new similar queries

The extraction of great amounts of information is a complicated task that depends upon the source where the information is stored. Each source presents several characteristics that determine the way of acquiring information and consequently these characteristics condition the model followed by extracting the available information. This work is focused on the Web and here search engines are the most important access points that exist for retrieving information. Thus the extraction process is dependent on the characteristics of search engines.

Among the characteristics of search engines that we can highlight are the small number of terms that a query contains and the ambiguity derived from this fact. Search engines do not allow a great number of terms for each query then works like the proposed one by Harman [3], where thousands of terms are added to the original query in order to maximize the effectiveness of the search process, cannot be implemented. Hence every strategy that tries to extract knowledge by using the Web like information source has to consider that the addition of terms cannot be an applicable solution depending on the number of terms to be added. For that reason our model proposes the interchange of terms instead of the addition of terms although we do not discuss the advantages of the query expansion by the addition of terms as way of improving the search process.

This model proposes a definition for the user query according to several rules that will allow the generation and selection of the best reformulated queries for the search process. After defining the query, the system should generate a set of alternative queries by replacing each original query term by other term from a knowledge resource such as an ontology or a thesaurus. From the set of alternative queries, the system has to be capable of selecting the best queries because the previous process can generate thousands of queries. This decision is made by taking into account the constraints imposed by the user on the original query and the relationship degree between the original terms and the terms extracted from the knowledge resource used. The imposed constraints will have similar meaning to the imposed ones in other works [10], but in this case they will affect to the selection of reformulated queries, not to the selection of documents. The final result is a set of queries that will be used to retrieve information from the Web. Hence our model consists of the following steps that are described in detail in the next sections: (1) Query definition, (2) Reformulation and (3) Filtering.

3.1 Query definition

According to Bordogna [14] a fuzzy linguistic-weighted Boolean query with associated semantics can be defined as a Boolean expression whose atomic components are pairs $\langle t_i, w_i \rangle$ belonging to the set, $T \times H$ (Importance); t_i is an element of the set T of terms, and $w_i \in [0,1]$ is a value of the linguistic variable, Importance, with qualifying the importance that the term t_i must have in the desired documents. This query definition allows the insertion of weights to the query terms in order to measure the importance of each term [13]. In our case we extend that definition by adding a new weight w_i^2 that represents the threshold that the user demands to consider a reformulated concept as acceptable and we change the meaning of the weight w_i^1 that represents the importance of the concept t_i into the boolean expression where it is defined w.r.t. to the other concepts of that boolean expression. Besides we define a new weight w_i^3 that represents the similarity degree between an original concept and a reformulated one. Therefore a query can be defined by a boolean expression whose atomic terms are 4-tuple $\langle t_i, w_i^1, w_i^2, w_i^3 \rangle$ belonging to the set TxS^2 where $t_i \in T$. Both w_i^1 and w_i^2 are ordinal values of the linguistic variable *Importance* modeling the threshold semantics and the importance semantics. Furthermore both weights can be assigned to each boolean expression, i.e., we are not working with terms, we are working with concepts. On the contrary w_i^3 is a numerical value that can be transformed into a label by the function called "Label" which assigns a label in S to a numerical value $r \in [0,1]$. That function is defined according to the following expression [15]:

$$Label(r) = Sup_q\{s_q \in S : \mu_{s_q}(r) = Sup_v\{\mu_{s_v}(r)\}\} \quad (1)$$

Thus we can denote by R the set of all legitimate queries that can be built following these rules:

1) $\forall \langle t, w^1, w^2, w^3 \rangle \in Tx\,[0,1] \rightarrow \langle t, w^1, w^2, w^3 \rangle \in R$.
2) $\forall q_1, q_2, \ldots, q_n \in R$ and $w^1, w^2, w^3 \in [0,1] \rightarrow \langle q_1 \wedge q_2 \wedge \ldots \wedge q_n, w^1, w^2, w^3 \rangle \in R$.
3) $\forall q_1, q_2, \ldots, q_n \in R$ and $w^1, w^2, w^3 \in [0,1] \rightarrow \langle q_1 \vee q_2 \vee \ldots \vee q_n, w^1, w^2, w^3 \rangle \in R$.
4) $\forall q_1 \in R \rightarrow \neg q_1 \in R$.
5) All legitimate boolean queries $q \in R$ are only those obtained by applying rules $1-4$ inclusive. The weights of the rules 2 and 3 are not relevant for the reformulated queries, only for the original query.

Each query defined by the preceding rules represents a concept. We can differentiate two kinds of concepts: atomic and compound. An atomic concept is formed by a single term following the rule 1 and a compound concept is a query that consists of logic operators following the rest of the rules.

In order to operate with a query we need to extend the query definition again. A query is defined by the specification of a concept that the user desires satisfied. The user defines his query q by expressing it in terms of a group of concepts c_j, an importance weight associated with each of the components, w^1, a threshold semantics associated with each of the components, w^2, a similarity semantics associated with each of the components, w^3, and a quantifier ϑ is also necessary which will indicate a mode of interaction between the query components [16]:

$$q = \langle C_1, C_2, \ldots, C_n : w^1 : w^2 : w^3 : \vartheta \rangle$$

Remark 1: It is necessary to note that the linguistic quantifier ϑ has no sense for atomic concepts, for that reason, it is not necessary to define it in that case.

For each component i from the query that is not an atomic concept the user is asked to provide a definition, that is, the aggregation operator ϑ_i to be used and the weights w_i^1, w_i^2, w_i^3 for the reformulation process. This process begins with the atomic concepts and is continued until

the complete hierarchy defining the query is formulated. It is noted that this hierarchy is a tree like structure in which the leaves are atomic components. The specification of the importance and the threshold weights is optional. If those weights are not expressed, then by default they are assumed to have an importance value of one. The similarity weight for each component from the original query is one, and for the reformulated queries depends on the resource used for exchanging the query terms. In addition to the weights defining the user need, a set of linguistic quantifiers, $\vartheta = \{\vartheta_1, \vartheta_2, \ldots, \vartheta_q\}$ available to the user is necessary. The user is who has to select the most suitable quantifier but the system can designate one of them as the default quantifier whether the user does not choose one of them. A transformation function *h* and a aggregation operator of linguistic information *f* are transparent to the user in order to aggregate linguistic weighted information [11].

Moreover the user has to set the minimum resemblance between the original query and each reformulated query. This constraint is defined by the threshold ψ which will determine if a reformulated query is accepted or rejected for the search process. If the result of the evaluation of the whole query (see section 3.3) is greater than ψ then the query is accepted, otherwise the query will be rejected.

Following the idea of Importance proposed by Bordogna [14] and modeled by a fuzzy linguistic approach by Herrera-Viedma [10] we model the semantics of our method by a set of ordinal linguistic values S w.r.t. the linguistic variable Importance. These semantics are explained in the following subsections.

3.1.1 Importance semantics (w^1).

This semantics measures the relative importance of each component w.r.t. the components of the concept in which is included. This semantics will be used to calculate the similarity degree of each reformulated subconcept w.r.t. the original one in order to evaluate the importance degree of a reformulated concept. For this calculation it is necessary to aggregate the weights w^1 and w^3 of the sons of the tree that represents the boolean expression that is being evaluated, thus, given the boolean expression shown in fig. 1, the vectors $M^1 = \{L, M, H\}$ and $M^3 = \{0.4, 0.5, 0.3\}$ will be aggregated in order to obtain a final value that will represent the similarity between two queries.

If that final value is not greater than the weight assigned to the threshold weight w^2 then the reformulated query is considered as non-useful for the search process because is not enough similar w.r.t. the original one. Its use will be explained in subsection 3.3.

Remark 2: The importance semantics of a whole query has no sense because this query cannot be compared with others. This semantics has sense only for subexpressions.

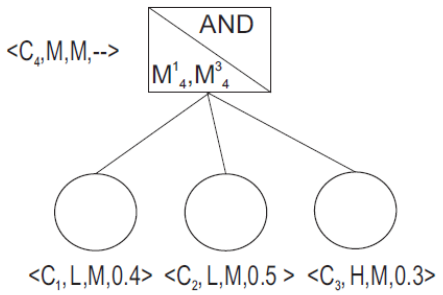


Fig. 1: Reformulated boolean subexpression

3.1.2 Threshold semantics (w^2).

The threshold value is an order linguistic value assigned by a user whose objective is to describe the minimum similarity degree that can be assumed in order to consider a reformulated query as valid.

Considering a linguistic approach this weight w^2 can be interpreted as a value of the linguistic variable *Importance* [14], [10]. The primary term adopted is "important". The linguistic weights are variations of the primary term important, e.g., at least very important, at least averagely important, at least minimally important, etc. In this model a concept $\langle C_i, -, w_i^2, - \rangle$ is synonymous with the concept $\langle C_i, -,$ at least $w_i^2, - \rangle$, and therefore, the request expresses the minimally acceptable similarity degree between an original concept and a reformulated concept.

Thus given an original concept C_{oi} and a reformulated concept C_{ri}, if the similarity degree between both is "Low" and the user set a threshold value "Medium" ($\langle C_{oi}, -, Medium, - \rangle$) then the reformulated concept C_{ri} should be rejected as well as the reformulated query in which the concept C_{ri} is defined. This semantics is inspired in other works related to document retrieval [10].

This semantics is assigned by the user in the query initialization. If the user does not assign a value then the system can assign a value by default.

Remark 3: The reformulated queries preserve these values from the original one for the filtering process which will select the most suitable queries. Thus a concept defined as $\langle C_{oi}, -, Medium, - \rangle$ is redefined by other concept $\langle C_{ri}, -, Medium, - \rangle$ where the threshold value ('Medium') is the same in both concepts.

Remark 4: The threshold ψ defined by the user is the same threshold for the whole query.

3.1.3 Similarity semantics (w^3).

The weight w^3 indicates a relationship degree between the original concept and the reformulated one. In order to consider a concept as useful for the search process this weight has to exceed its corresponding weight w^2. The similarity weights of the query atoms can be assigned by an ontology or a thesaurus for instance, but the weight of the subconcepts

represented by boolean expressions is calculated during the filtering process (see section 3.3) by the aggregation of the semantics weights of the underlying subconcepts and their corresponding importance weights. For that purpose the linguistic aggregation operators commented in Definition 1 and 2 can be used whereas the numerical values can be transformed into linguistic values by using the equation 1.

Remark 5: Obviously the similarity between the original query and itself is maximum, therefore all the components of the original query weigh one for the similarity semantics.

3.1.4 Example.

In order to simplify the example consider a set of labels consisting of only 5 label:

$$S = \{S_0 = None, S_1 = Low, \quad S_2 = Medium, S_3 = High, \\ S_4 = Perfect\}.$$

and the original query q_o with its mathematical representation:

$$q_o = \{\langle c_{o1}, L, M, 1\rangle, \langle c_{o2}, H, M, 1\rangle, \langle c_{o3}, M, M, 1\rangle, \\ \langle c_{o4}, M, M, 1\rangle, \langle c_{o5}, H, M, 1\rangle, \langle c_{o6}, M, M, 1\rangle, \\ \langle c_{o7}, L, M, 1\rangle, \langle c_{o8}, -, M, 1\rangle\}.$$

As can be seen the previous formula represents an ordered set consisting of the subconcepts that compose the original query. The compound concepts are defined as:

$$c_{o6} = \langle c_{o1} \text{ AND } c_{o2} \text{ AND } c_{o3} : M : M : 1 : \vartheta_6 = LWC\rangle.$$
$$c_{o7} = \langle c_{o4} \text{ OR } c_{o5} : L : M : 1 : \vartheta_7 = LWD\rangle.$$
$$c_{o8} = \langle c_{o6} \text{ OR } c_{o7} : - : M : 1 : \vartheta_8 = LWD\rangle.$$

While the importance vectors are:

$$M_6^1 = \{L, H, M\}, M_7^1 = \{M, H\}, M_8^1 = \{M, L\}.$$

A value Medium (M) has been assigned as threshold to every concept in order to demand a minimum resemblance between the original concepts and the reformulated ones. Therefore the threshold for the whole query is $\psi = M$.

3.2 Reformulation

The reformulation of the original query consists in interchanging some or all the atomic terms by replacing them with new terms from a knowledge source that allows calculating the similarity degree between both the original one and the reformulated one. The idea of interchanging allows avoiding the growth of the query because if we want to apply this idea on Web search engines then it is necessary to take into account that the length of the queries is limited; we cannot create queries larger than a few terms.

The knowledge source used for obtaining related terms has to provide a similarity degree that will be used for the filtering process. The similarity degree is considered as a fuzzy weight that has to be normalized in the interval $[0, 1]$. Thus this source has to establish a relation F between two terms $t_1, t_2 \in T$ such that $F : TxT \rightarrow [0, 1]$. For example a thesaurus can be an interesting source due to its two basic properties for two terms t_1, t_2: (i) identity, $T(t_1, t_2) = 1$ for all $t_i \in T$, and (ii) symmetry, $T(t_1, t_2) = T(t_2, t_1)$.

The reformulated query preserves all the parameters from the original query except the similarity degree because is the unique parameter assigned automatically, the others are defined by the user.

The final result of this stage is a set of reformulated queries. The cardinality of this set depends on the possible combinations that can be performed with the terms of the knowledge source. That cardinality can be very large and for that reason it is necessary to filter some of the queries selecting only the best ones.

Remark 6: If one term is not interchanged then its similarity degree is one as well because it is an original term and we assume that the user query is the best definition of his information request.

3.2.1 Example of reformulated query.

When a new query is reformulated then the underlying concepts are preserved but several of the new terms are different w.r.t. the original query.

Analyzing the reformulated query q_r

$$q_r = \{\langle C_{r1}, L, M, 0.7\rangle, \langle C_{r2}, H, M, 0.8\rangle, \langle C_{r3}, M, M, 0.9\rangle, \\ \langle C_{r4}, M, M, 1.0\rangle, \langle C_{r5}, H, M, 0.8\rangle, \langle C_{r6}, M, M, -\rangle, \\ \langle C_{r7}, L, M, -\rangle, \langle C_{r8} -, M, -\rangle\}.$$

As can be observed the atoms have a numerical weight assigned through a thesaurus for example, whereas the rest of values are empty because have to be calculated by using the corresponding linguistic aggregation operator. Thus the unique parameters that have been modified w.r.t. the original query are the similarity weights of the atomic concepts:

$$M_6^3 = \{0.7, 0.8, 0.9\}, M_7^3 = \{1, 0.8\}$$

The value M_8^3 is calculated by aggregating the values from M_6^1, M_7^1 and M_6^3, M_7^3 as is explained in the following subsection.

3.3 Filtering

The aim of this model is to calculate the relationship degree between the original query q_o defined by the user and a reformulated query q_r with its associated quantifiers ϑ and M^1, M^2, M^3 by a matching function D in order to compare that result with a threshold ψ which will decide if the query q_r is accepted for the search process or not:

$$D(q_r) = F_{\vartheta/M^1 M^2 M^3}(C_1(q_r), C_2(q_r), \ldots, C_n(q_r)).$$

In order to evaluate the similarity of the query q_r it is necessary to evaluate each subconcept C_i, starting with the atoms and following with the boolean expressions. The process finishes when the whole query has been evaluated. According to Waller [17]: "by separability is meant the ability to a query by evaluating first the individual terms, then Boolean combinations of the terms, and so forth, working in a bottom-up fashion". Therefore we follow this idea in our evaluation process.

For a given query, the filtering process acts as a hierarchical process distinguishing three evaluation levels: (i) evaluation of individual atoms, (ii) evaluation of Boolean subexpressions, and (iii) evaluation of the whole query. In the first level the process compares the value of the threshold weight w^2 and the similarity weight w^3. If $w^3 \geq w^2$ then the process follows with the rest of concepts, otherwise the query is rejected as useful for the search process. In the other two levels the weight w^3 is calculated through the vectors M^1 and M^3. The aggregation of the pairs $\langle M_i^1, M_i^3 \rangle$ of a subconcept i is computed through the aggregation operators.

The weighted logical connectives AND and OR are modeled by means of i.e the aggregation operators of linguistic weighted information LWC and LWD, respectively. We should note that these operators guarantee the correct application of the importance semantics because both use transformation functions that try to reduce the effect of elements with low importance in the resulting aggregated information. To do so, in the first operator, the elements with low importance are transformed into small values and in the second one into large values [11], [10]. If this aggregation step returns a value "High", this fact indicates that the capability of representing the concept C_{oi} through the reformulated concept C_{ri} is "High". This result of the aggregation has to be compared with w_i^2 in order to decided if the analyzed concept is useful or not.

Consider the query q_j which can be represented as follows:

$$q_j = \{C_1 = t_1, C_2 = t_2, C_3 = t_3, C_4 = t_4, C_5 = t_5, C_6, C_7, C_8\}.$$

The first elements of the set that represents the query q_j are the query terms $(C_1, C_2, \ldots)$ and the last element (C_n) is the root of the tree, i.e., represents the whole query. The final result of the filtering process is a fuzzy subset characterized by the membership function:

$$\mu_{q_j} : R \rightarrow S, \text{ i.e., } q_j = \sum_{i=1}^{n} \mu_{q_j}(C_i)/C_i.$$

μ_{q_j} is the relationship degree between each subconcept from the reformulated query q_r and the original one, and the evaluation of the whole query is the value $\mu_{q_j}(C_n)/C_n$. The relationship degree (μ_{C_i}) for each subconcept C_i depends on whether that concept is a boolean expression or a term as was commented before. If $\mu_{q_j}(C_n) = None$ then the query represented by C_n is not accepted for the search process.

The previous formula is related to one individual query but the final result of the filtering process is set of queries that can be interpreted as a fuzzy subset P which contains the most important queries from the reformulated set R that can be submitted to each search engine in order to extract information from the Web:

$$M(R) = \sum_{i=1}^{n} \mu_M(q_i)/q_i.$$

In the algorithm 1 the process for generating the final set of queries P is described.

Algorithm 1 Calculation of the most similar queries

Require: $R = \{q_1, q_2, \ldots, q_n\}$, ψ.
1: $w \Leftarrow 0$, $P = \emptyset$.
2: **for all** $q_l = \{C_i | i = 1 \ldots n\} \in R$ **do**
3: $i \Leftarrow 1$, $accept \Leftarrow true$
4: **while** $i \leq n$ AND $accept = true$ **do**
5: **if** C_i is a term **then**
6: $w \Leftarrow w_i^3$
7: **else**
8: $w \Leftarrow aggregate(M_i^1, M_i^3, \vartheta_i)$
9: **end if**
10: $\gamma \Leftarrow Label(w)$
11: **if** $index(\gamma) \leq index(w_i^2)$ **then**
12: $accept \Leftarrow false$
13: $\gamma \Leftarrow None$
14: **end if**
15: $i \Leftarrow i + 1$
16: **end while**
17: **if** $index(\gamma) > index(\psi)$ **then**
18: $add(q_l, P)$
19: **end if**
20: **end for**
21: **return** P

The function *index* retrieves the index i of the $labelS_i$ and the function *add* incorporates a new query q_l into the set P.

4. Example

Suppose the original query q_o defined by a user and a thesaurus that helps us to reformulate that query into a new query q_r. The first step that has to be carried out is to check if the similarity weights w_r^3 of the atomic concepts are greater than their corresponding threshold weight w_r^2. We suppose that the numerical values are transformed into linguistic values by using the equation 1: $Label(0.7) = H$, $Label(0.8) = H$, $Label(0.9) = H$, $Label(1) = P$. As the similarity semantics of all the atoms exceed the threshold semantics:

$$\begin{aligned} C_{r1} &\Rightarrow \text{ Label}(0.7) > M \Rightarrow H > M, \\ C_{r2} &\Rightarrow \text{ Label}(0.8) > M \Rightarrow H > M, \\ C_{r3} &\Rightarrow \text{ Label}(0.9) > M \Rightarrow H > M, \\ C_{r4} &\Rightarrow \text{ Label}(1.0) > M \Rightarrow P > M, \\ C_{r5} &\Rightarrow \text{ Label}(0.8) > M \Rightarrow H > M, \end{aligned}$$

then the similarity degrees w_6^3, w_7^3, w_8^3, to the corresponding concepts C_6, C_7, C_8 can be calculated:

$$w_6^3 = LWD((Label(0.7), L), (Label(0.8), H), (Label(0.9), M)) = LWD((H, L), (H, H), (H, M)).$$
$$w_7^3 = LWC((Label(1), H), (Label(0.8), H)) = LWC((P, H), (H, H)).$$
$$w_8^3 = LWC(a_1, a_2) = LWC(a_1, a_2).$$

Fixing the transformation function of LWD as $h = LC_1^{\rightarrow} = MIN$ and the transformation function of LWD as $h = LI_1^{\rightarrow} = MAX(Neg(a), b)$ we obtain the following subexpression evaluations:

$$w_6^3 = LWD(H, L), (H, H), (H, M) = MAX\{MIN(H, L), MIN(H, H), MIN(H, M)\} = H.$$
$$w_7^3 = LWC((P, H), (H, H)) = MIN\{MAX(Neg(P), H), MAX(Neg(H), H)\} = H.$$

In this case the similarity semantics of all the boolean subexpressions exceed the threshold semantics again: $C_{r6} \Rightarrow H > M$; $C_{r7} \Rightarrow H > M$, and consequently the whole query can be evaluated: $w_8^3 = LWC(H, H) = MAX(Neg(H), H) = H$. The final value of the evaluation of the query q_r is 'High' and the threshold ψ fixed by the user is 'Medium', therefore that query is considered as interesting for the search process.

5. Conclusions

This work presents a model for generating related queries that can be useful for extracting information from the Web. From a user query, the system is able to generate a huge set of queries through the use of a knowledge source. The query definition allows the user to insert many constraints to his query by means of the assignment of linguistic values to the different subconcepts of the query. The set of generated queries has to be filtered in order to extract only the most useful for the search process. The filtering process is based on a bottom-up algorithm which analyzes each subconcept that composes the query. This algorithm processes the reformulated query aggregating the relationship and importance weights fixed by the user and checking if the final result of this aggregation is larger than the thresholds fixed for each subconcept. If the answer is affirmative then the query will used for the search process, otherwise the query will be rejected.

Acknowledgments.

Partially supported by FIDELIO (TIN2010-20395) supported by Science and Innovation Ministry, FOG (TSI-020302-2009-55) supported by Industry, Turism and Trade Ministry, SCAIWEB-2 excellence project (PEIC09-0196-3018) and PLINIO (POII10-0133-3516) supported by Education & Science Department of Regional Goverment of CastillaLa Mancha (Spain)

References

[1] B. J. Jansen, D. L. Booth, and A. Spink, "Patterns of query reformulation during Web searching," *Journal of the American Society for Information Science and Technology*, vol. 60, no. 7, pp. 1358–1371, 2009. [Online]. Available: http://dx.doi.org/10.1002/asi.21071

[2] E. N. Efthimiadis, "A user-centred evaluation of ranking algorithms for interactive query expansion," in *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, 1993, pp. 146–159.

[3] D. Harman, D. Benson, L. Fitzpatrick, R. Huntzinger, and C. Goldstein, "IRX: an information retrieval system for experimentation and user applications," *SIGIR Forum*, vol. 22, pp. 2–10, 1988.

[4] R. Jones, B. Rey, O. Madani, and W. Greiner, "Generating query substitutions," in *WWW '06: Proceedings of the 15th international conference on World Wide Web*, New York, NY, USA, 2006, pp. 387–396.

[5] B. J. Jansen and A. Spink, "An analysis of web searching by European AlltheWeb.com users," *Information Processing and Management*, vol. 41, pp. 361–381, 2005.

[6] G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais, "The vocabulary problem in human-system communication," *Communications of the ACM*, vol. 30, pp. 964–971, 1987.

[7] P. Bruza and S. Dennis, "Query Reformulation on the Internet: Empirical Data and the Hyperindex Search Engine," in *Proceedings of RIAO '97 - Computer Assisted Information Searching on the Internet*, Montreal, Canada, 1997, pp. 500–509. [Online]. Available: http://www.workingweb.com.au/training/RIAO97.pdf

[8] S. Y. Rieh and H. Xie, "Patterns and sequences of multiple query reformulations in Web searching: A preliminary study," in *Proceedings of ASIST Annual Meeting, Washington DC, Nov 2001*, 2001, pp. 246–255.

[9] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning," *Information Sciences*, vol. 8, no. 3, pp. 199–249, 1975. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/0020025575900365

[10] E. Herrera-Viedma, "Modeling the retrieval process for an information retrieval system using an ordinal fuzzy linguistic approach," *Journal of the American Society for Information Science and Technology*, vol. 52, no. 6, pp. 460–475, 2001.

[11] F. Herrera and E. Herrera-Viedma, "Aggregation operators for linguistic weighted information," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 27, no. 5, pp. 646–656, 1997. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=618263

[12] ——, "Linguistic decision analysis: steps for solving decision problems under linguistic information," *Fuzzy Sets and Systems*, vol. 115, no. 1, pp. 67 – 82, 2000. [Online]. Available: http://portal.acm.org/citation.cfm?id=351884

[13] R. R. Yager, "A note on weighted queries in information retrieval systems," *Journal of the American Society for Information Science*, vol. 38, no. 1, pp. 23–24, 1987. [Online]. Available: http://doi.wiley.com/10.1002/(SICI)1097-4571(198701)38:1<23::AID-ASI4>3.0.CO;2-3

[14] G. Bordogna and G. Pasi, "A fuzzy linguistic approach generalizing boolean information retrieval: a model and its evaluation," *Journal of the American Society for Information Science*, vol. 44, pp. 70–82, 1993.

[15] F. Herrera, E. Herrera-Viedma, and J. L. Verdegay, "A model of consensus in group decision making under linguistic assessments," *Fuzzy Sets and Systems*, vol. 78, no. 1, pp. 73 – 87, 1996. [Online]. Available: http://portal.acm.org/citation.cfm?id=229505

[16] R. R. Yager, "A Hierarchical Document Retrieval Language," *Information Retrieval*, vol. 3, no. 4, pp. 357 – 377, 2000. [Online]. Available: http://www.springerlink.com/content/x3mv16j33847tvh6

[17] W. Waller and D. H. Kraft, "A mathematical model of a weighted boolean retrieval system," *Information Processing and Management*, vol. 15, no. 5, pp. 235–245, 1979. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/030645737990030X

From Text Documents to Causal Mechanisms

Cristina Puente[1], **José A. Olivas**[2] , **Roberto Merlo**[1]
[1] Advanced Technical Faculty of Engineering – ICAI, Pontificia Comillas University, Madrid, Spain
[2] Information Technologies and Systems Dept., University of Castilla-La Mancha Ciudad Real, Spain

Abstract –*In this paper we explore the role of causality when certain linguistic structures, detected automatically by a program suited for this task, are present into a sentence. We have developed two algorithms, one to extract and analyze static texts, like books or documents, and another one which deals with dynamic information and collects and filters HTML code from different web sites related to a topic.*

Keywords: Causal relationships, Web search, Soft-computing, deduction.

1 Introduction

Causality plays and has played an important role in human cognition, in particular in human decision-making, providing a basis for choosing an action which is likely to lead to a desired result. There are many works and theories about this theme, philosophers, scientists, physics, mathematicians, computer scientist and many others have explored the field of causation starting with the ancient Greeks three thousand years ago.

Causation plays a different role when analysed from different fields. In legal ambiences it is a matter of conduct and result, while in science, such physics it is the result of empirical experiments and evidence.

In daily life, causality is present in many situations. If someone fails to stop at a red light and there is a car accident, it can be said that the failure to stop was the cause of the accident. However, failing to stop at a red light does not guarantee that an accident will happen for sure [1]. Sometimes, true statements do not lead to a valid reasoning, as in J. Pearl's example [2]:

1. *"If the grass is wet, then it rained.*
2. *If we break this bottle, the grass will get wet.*
3. *Output → If we break this bottle, then it rained".*

Causality is not only a property of physical facts. Social events are also causally linked [3]. So, we say that '*poverty causes ignorance*' or perhaps that '*ignorance causes poverty*'. Social causality see causal links not as an uncertainty process, but vague or poorly defined. While physics deals with a well-defined universe of discourse, social sciences refer to frequently ill-defined, incomplete or inaccurate collections.

Under these conditions, it is reasonable to say that causality is an imprecise and imperfect relationship between two entities, cause and effect. For this reason, Zadeh [4] remarks that does not exist a definition of causality within the conceptual structure of classical logic or probability theory, able to provide a reasonable answer to the following points:

- The definition has to be general, not restricted to a narrow class of systems or phenomena.
- This definition has to be precise and unambiguous, in order to be used as basis for logical reasoning or computation.
- Given two causally connected events, *A* and *B*, this definition has to be able to answer these questions:
  - Did or does or will *A* cause *B* or vice-versa?
  - If there is a causal link between *A* and *B*, what is the strength?

Zadeh also points out three sources of difficulty in defining or establishing causality. The first one is chaining, that is a temporal chain of events, $A_1, A_2, A_3 \ldots A_n$, which ends on A_n. The difficulty here relies on determining to what degree (if any) A_i(i=1,….,n-1) causes A_n. In [1], it is introduced some examples about the problems that causal chains present:

- Simultaneous Plant Death: "*my rose bushes and my neighbour's rose bushes both die. Did the death of one cause the other to die? (Probably not, although the deaths are associated)*".
- Drought: "*There has been a drought. My rose bushes and my neighbour's rose bushes both die.*

Did the drought cause both rose bushes to die? (Most likely)".

- Traffic: "*A friend of mine calls me up on the telephone and asks me to drive over and visit him. While driving over, I ignore a stop sign and drive through an intersection. Another driver hits me, and I die. Who caused my death?*"

- Poisson: "*Fred and Ted both want Jack death. Fred poisons Jack's soup, and Ted poisons his coffee. Each act increases Jack's chances of dying. Jack eats the soup, and feeling rather unwell leaves the coffee, and dies later. Ted's act raised the chance of Jack's death but was not cause of it*".

Another problem that Zadeh remarks is the confluence or conjunction. In this case it is presented a confluence of events $A_1,A_2,A_3….A_n$ and a resultant event *B*. the problem here is to calculate to what degree each event separately caused the final event B. To demonstrate this problem, Zadeh proposes two examples:

- *A raincoat manufacturer would like to increase his sales. To this end, he increases the advertising budget by 20%. Six months later, sales went up 10%. Was the increase on sales caused by the increase in the advertising budget? If so, to what degree?*

- *Business news announces that the stock market had a sharp drop. Analysts cite as primary reasons for the drop a 2% increase in unemployment, and 3 dollar-a-barrel increase in the price of oil. To what degrees did the unemployment and the price of oil caused the sharp drop?*

The third problem is covariability, seen as a statistical association. In this case, *A* and *B* are variables, and there appears to be a deterministic or statistical covariability between *A* and *B*. The problem is establishing if this covariability is a causal relation. Moreover, when is a relation a causal relation? Differentiation between covariability and causality presents a difficult problem, especially in the context of data mining. Causality is referred to demonstrated and established facts; given a cause, an effect happens, but covariability is related more to coincidence and undemonstrated set of facts, despite it is noticeable that if something is changed (the cause), the output is affected (effect). For example, it is generally assumed that aging causes a loss in acuity of hearing. However, recent studies have shown that the loss in acuity is caused by prolonged exposure to high levels of sound and not by aging per se. Or another example, *I fell at home and broke my right leg and my left arm. Is there a causal connection between breaking my right leg and left arm?*

Causal statements usually have three properties which mean that the cause must precede the effect, cause and effect must be materially related, and whenever the cause happens, the effect must take place [5]:

- Asymmetry: the cause always happens before the effect.

- Linearity: any cause is followed by an effect.

- Transitivity: if *A* causes *B* and *B* causes *C*, *A* causes *C*.

Taking all these premises into account, in this work we have explored the role of causality in text documents and in web pages. So it is first described an algorithm to extract and classify conditional and causal sentences from text documents. This algorithm is based on 20 causal patterns. As complement, we have developed another process to locate a site and search for a given topic as well as the pages where this topic is present. This combination will serve us to compose both a dynamic and static base of knowledge.

2 Conditional automatic detection

One of the main difficulties that we found was how updated the information is. Nowadays, the Internet is a huge source for information, so it has to be taken into account in order to be documented about an issue. For example, in the case of the virus Influenza, its treatment, symptoms, etc, the Internet was collapsed of useful information to study the illness. Many times, if the information is recent in time, there are not reliable books about it, so it raises the need to use Internet to be updated.

In the process described in [6], as seen in figure 1, we presented an algorithm to detect conditional and causal statements from text documents. This process analyzed large documents or books about an issue to extract causal knowledge producing as a result a causal graph related to a topic.

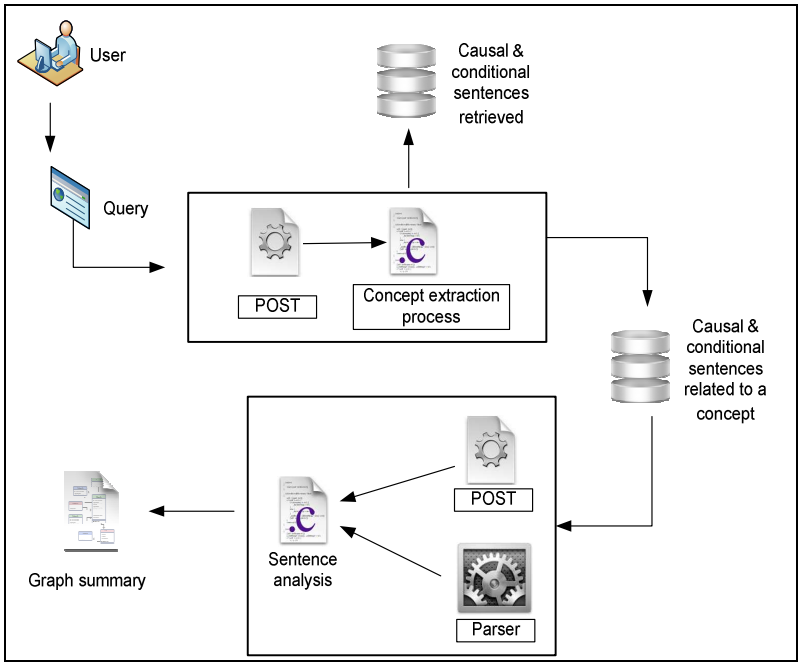


Fig. 1 Procedure to extract and analyze causal sentences, and produce a causal graph.

To do so, we have first to divide a phrase into its basic components (verbal tenses, adverbs, pronouns, nouns, etc.) in order to develop an algorithm able to detect and classify conditional and causal sentences based on structured patterns. According to the English grammar [7], there are two types of lexicon that characterize conditional sentences: (i) some selected verbs (ii) some discourse makers. The phrases are extracted in accordance with 20 linguistic patterns which cover the first, second and third conditional in English, or causative phrases including causal particles as due to, provoked by, provided that, have something to do with, .etc.

We used a morphological analyzer Flex1 plus the C programming language to create a six-state automaton able to recognize these patters and create a file with the obtained results.

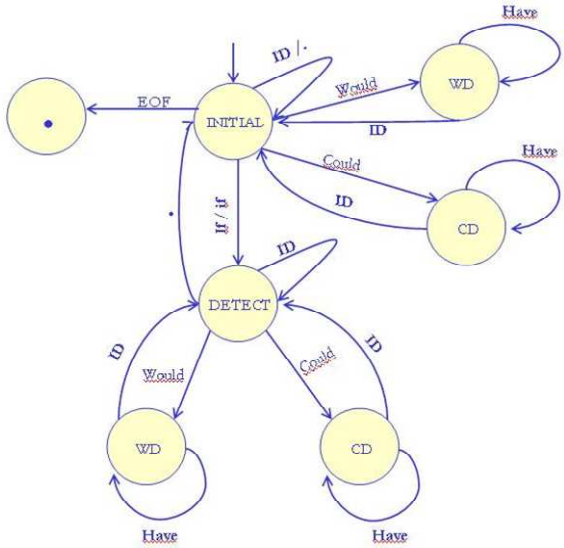


Fig. 2 Automaton to detect and classify causal sentences.

In order to perform a 'gold standard' for evaluation, some documents belonging to different text categories were manually analyzed (50 pages per category), and recall, precision and F-measure were calculated as seen in figure 3. Recall or R, is the number of correct causal sentences classified by the system divided by the number of causal

1 http://www.gnu.org.

2 http://www.nlm.nih.gov/bsd/mmshome.html

3 http://www.daviddlewis.com/

sentences classified by manual analysis. Precision or P, is the number of correct causal sentences classified by the program, divided by the total amount of sentences retrieved. F-measure is a combination of recall and precision, using the formula F = (2*P*R) / P+R, to control the relative importance of recall and precision:

Type of text	Recall	Precision	F-Measure
Scientific	0,65	0,84	0,73
Medical (Medlars)[2]	0,77	0,91	0,83
Novels	0,32	0,55	0,41
News (Reuters)[3]	0,58	0,79	0,67
Gospel	0,50	0,70	0,58

Fig. 3 Recall, Precision and F-Measure of the analyzed texts.

These results show better performance with medical texts, with a recall factor of 83% and the highest value for precision 91%, and scientific texts than in general purpose texts (novels), Gospel texts and the news, where language is not as direct and concise.

3 Dynamic retrieval of information

The sources of information are vital in order to have an updated set of documents about a given theme. In this work we have used some relevant documents and books about the studied topics (for example in medicine, to study lung cancer), but some of these books were written a few years ago, so they may not be as updated as we would like.

To improve the collection of documents to be processed, and compose a static and a dynamic database of contents (static with the documents obtained from books, and dynamic with the information contained in web sites), we have designed an algorithm able to gather the textual information contained in a web page, filtering HTML tags and markers that are not part of the information.

The schema of this algorithm is presented in figure 4. The first step is to obtain an URL to get the information from. Given the good results of medical tests, we decide to study the relationship between *Smoking* and *Lung Cancer*, and look for these two topics within the Mayo Clinic web site.

In addition, we have programmed a mini-crawler capable of following the URL appearing in one of these sites. For example, if the URL to follow is the Mayo Clinic's, and inside that web page there is another URL pointing at another page located in this same domain, the crawler would understand that in that address there might be relevant information related to the topic, so it will get the information from this page as well. To avoid infinite loops, the crawler would only trace those URLs belonging to the same domain.

The goal of the web pages text retrieval module is to transform HTML code from a requested web page into text with relevant content. Each paragraph has to be lexically connected, so HTML tags will be evaluated for this task, separating different types of content (pictures, links, sections, etc.). On the other hand, there are some HTML tags used to format text, like This is a text in strong, that will have to be removed.

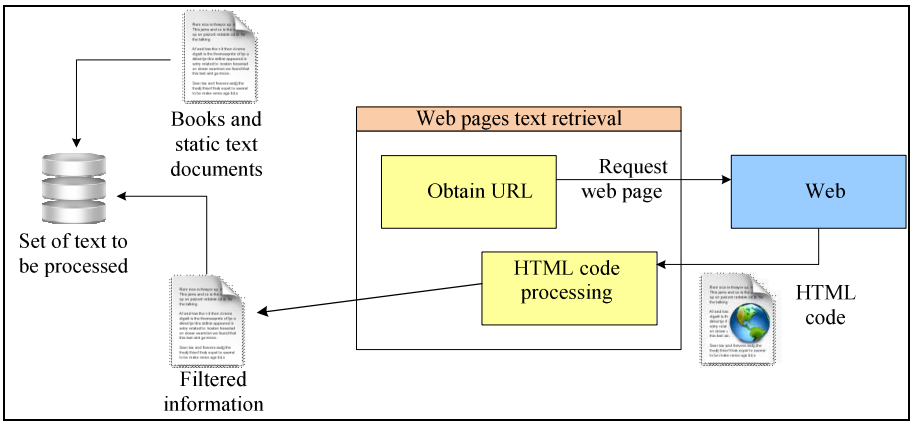


Fig. 4. Dynamic retrieval of information.

With these considerations, the algorithm to separate relevant text from a web page is resumed in the following points:

1) Set all the HTML tags to lower-case letters (case insensitive), otherwise the tags y <EM> would be considered as different. To do so, it has been transformed all the text contained between the symbols lower than (<) and greater tan (>), which are the tags delimiters.

2) Change all the HTML Entities to its correspondent character in plain text, according to the Latin-1 (ISO 8859-1) codification. For example, the entity " would be transformed into the character ", or the entity   would be a blank space.

3) Clean the page of invisible content, like comments (delimited by tags <!-- y -->), scripting code (usually JavaScript code located between <script> y </script> tags), and CSS embedded styles.

4) This step is highly important in order to obtain understandable text paragraphs (especially to be processed by a morphological analyzer as will be done in the next point). All the HTML tags defined to format an output text in a web page have to be removed. According to the standards HTML 4.01 / XHTML 1.0 the set of tags to format a text are:

5) Remove hyperlinks tags like <a> and .

6) Transform the remaining HTML tags into the "<#>" tag, which will be used to separate bocks of text in a page.

As a result of the described algorithm, we will obtain 'clean' blocks of text with the information from the web pages processed as seen in figure 5. These paragraphs will be included in the database with the static information, obtained from sources that do not need a filtering process, like electronic books and documental databases. This set of dynamic and static text information will be the input for the detection, classification and analysis of causal sentences.

Subscribe to our Living with cancer newsletter to stay up to date on cancer topics.
****
Lung cancer is a type of cancer that begins in the lungs. Your lungs are two spongy organs in your chest that take in oxygen when you inhale and release carbon dioxide when you exhale.
****
Lung cancer is the leading cause of cancer deaths in the United States, among both men and women. Lung cancer claims more lives each year than colon, prostate, ovarian, lymph and breast cancers

Fig. 5. Example of text extracted from a web page

4 Conclusions

The process described in this paper introduces a new automatic model to retrieve conditional and causal sentences related to a given concept from text documents and from web pages. This model can be used to compose an updated knowledge base which can serve as the basis for future works, being improved to establish causal or conditional relationships among several concepts.

5 Acknoledgements

Partially supported by TIN2010-20395 FIDELIO project, MEC-FEDER, Spain, PEIC09-0196-3018 SCAIWEB-2 excellence project, JCCM, Spain and POII10-0133-3516 PLINIO project, JCCM, Spain..

6 References

[1] L. J. Mazlack, "Imperfect Causality". Fundamenta Informaticae IOS Press. Vol. 59, pp. 191-201, 2004.

[2] J. Pearl, "Causality, models, reasoning, and inference". Cambridge University Press, 2000.

[3] J. U. Obgu, "An etnographic of education in an urban neighborhood". N. Y., Academic Press, 1974.

[4] L. A. Zadeh, "Letter to the members of the BISC group". Proc. of the BISC Int. Workshop on Fuzzy Logic and the Internet, 2001.

[5] M. Bunge, "Causality: the place of the causal principle in modern science". Cambridge: Harvard University Press, 1959.

[6] C. Puente, A. Sobrino, J. A. Olivas, R. Merlo, Extraction, Analysis and Representation of Imperfect Conditional and Causal sentences by means of a Semi-Automatic Process. Proceedings IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2010). Barcelona , Spain , pp. 1423-1430, 2010.

[7] English grammar, http://www.britishcouncil.org/

Intentional tags in folksonomy based ranking systems

Pedro López-Juárez[1], **José A. Olivas**[2]
[1]Information Systems and and Communications Technologies Dept., Comillas Pontifical University, Madrid, Spain
[2]Information Technologies and Systems Dept. University of Castilla-La Mancha, Ciudad Real, Spain

Abstract— *In this paper, a proposal for a FolkRank algorithm improvement is presented. The guidelines are focused in the use of intentional tags co-occurrence, considered as fuzzy concepts, in order to improve the algorithm precision and recall measures when it is used in a recommender system.*

Keywords: Folksonomies, FolkRank, recommender systems, fuzzy logic.

1. Introduction

When Thomas Vander Wal coined the term *folksonomy* in 2004 to denominate the way in which users tag his topics in Compuserve forums, he could not imagine how much popular it will become. In present paper we introduce the basis of a FolkRank algorithm modification. First of all, we present what motivate us this work, then we show preliminary data set verification works. Next we explain the basis of FolkRank algorithm to finally introduce our changes to it.

2. Background

The term *folksonomy* born in 2004 when Thomas Vander Wal [5] answer the question "*... Is there a name for this kind of informal social classification?*" with the sentence "*... So the user-created bottom-up categorical structure development with an emergent thesaurus would become a Folksonomy?*.

He said that a *folksonomy* have three tenets. A tag, an object being tagged, and the identity of the person who tag. The third tenet is very important because this is a new feature that opens a new dimension, the social dimension. Social bookmarking tools like Delicious[1] or Flikr in the beginning, and many other tools later, offers to his users the possibility to use tags from other users, to tag his own bookmarks.

Golder and Huberman [6] make a study in Delicious. They find a correlation between a number of bookmarks used by a user and the number of tags he used. Also, they focus on the type of tags used in the system, and groups them in 7 categories: "*Identifying What (or Who) it is About. Identifying What it Is. Identifying Who Owns It. Refining Categories. Identifying Qualities or Characteristics. Self Reference. Task Organizing*".

Cameron Marlow et al [7] test 12 social bookmarking tools with an objective, to find in them the social dimension, and they find it in the relations between users of the same organization or people with common hobbies. This "*socio-technical design*" affects the information generated by themselves, so they propose two taxonomies focused on the design of this type of social bookmarking as a result of this analysis:

- *System design and attributes. We claim that the place of a tagging system in this taxonomy will greatly affect the nature and distribution of tags, and therefore the attributes of the information collected by the system.*
- *User incentives. User behaviors are largely dictated by the forms of contribution allowed and the personal and social motivations for adding input to the system. The place of a tagging system in this taxonomy will affect its overall characteristics and benefits.*

Andreas Hotho et al. [8][11] propose a folksonomy practical application, in search and ranking in folksonomy based systems. They propose a PageRank [3] based algorithm named **FolkRank**. First of all, they present a formal definition of folksonomies, and a definition of *personomy*.

We cite this works because they are the base of our work presented in this paper:

1) Vander Wal give us the three tenets of a folksonomy.
2) Golder and Huberman give us a tag categorization.
3) Marlow find a social dimension.
4) Finally, Hotho et al. develops the tool that we try to improve.

3. Data set verification

Since Yahoo! acquired Delicious, has become extremely difficult to obtain data about their users, because they limited to 500 the number of simultaneous bookmarks that can be retrieved. Then, we find CiteULike[2], a free service for

[1]On April, 28th, Yahoo! announced to Delicious users, that Delicious has been acquired by the founders of YouTube, and became part of their new Internet company, AVOS.

[2]http://www.citeulike.org

managing and discovering scholarly references.

3.1 The files

CiteULike offered free of charge to researchers 3 simple unix text files with type "\n" line breaks, and "|" character as a field separator. It should be emphasized the fact that the files are constantly growing [3], and data is available from May 30, 2007 onwards.

3.1.1 Who-posted-what data:

The file constitutes an anonymous dump of who posted what and when the posting took place. Contains 13.506.170 rows that collects all stored items from 4/11/2004 to 10/01/2011 and each contains:

1) The CiteULike article id which was posted.
2) A representation of the user name you put the item. It uses the encrypted username to preserve the identity of the users.
3) The date and time the article was posted to the site.
4) The tag the user used to catalog the article.

If a user posts an article with *n* tags, then this will result in *n* rows in the file.

3.1.2 Article linkout data:

The file allows to establish links between the articles identification numbers in CiteULike and its related resources on the web using 5 fields that allows it to compose a url. Contains 6.422.450 records, each containing:

1) The CiteULike article id.
2) Link type. It's a 6 characters field.
3) ikey_1.
4) ckey_1.
5) ikey_2.
6) ckey_2.

The *i* key (i = integer) and *c* key (c = character), contains information about how to build a standard URL[4]. The idea is that it tries to capture the internal identifier used by each system to represent the article.

3.1.3 Group membership data

The file constitutes an anonymous dump of *who* is a member of each *group*. Data is available from 2008-11-14 onwards. In CiteULike, groups are sets of users creating shared libraries links. They are very useful for tracking on a particular topic or about what a particular working group is reading. Same as *Who-posted-what data*, the file is a simple unix ("\n" line endings) text file with pipe ("|") delimiters. It contains 16.835 records with the following fields:

1) An obfuscated group identifier.
2) An obfuscated representation of the username (a encrypted username, in the same way as is done for the article posting data).

Same as *Who-posted-what data*, if a group has *n* members, then this will result in *n* rows in the file.

3.2 Preliminary data analysis

We known CiteULike is a tool with a more specific purpose than Delicious, that we consider a general purpose tool. Because this different purpose, it's necessary to verify CiteULike users behavior, and we took Marlow's work to compare the results.

After a preliminary data analysis, we found that 87.445 users have used 618.315 different tags to catalog 4.349.404 articles.

3.2.1 Tags

After this preliminary analysis, we obtain the result that the tags use frequency follow a falling power law, as is the case in all the works consulted[13]. Two of this tags are automatically generated by the system "*bibtex-import*" (253.426 times) and "*no-tag*" (549.549 times). The first tag appear because CiteUlike allow import BibTex bibliography references. The second is the default tag, the tag that system assigns when a user don't assign any tag to an article.

After extract tags and check them visually, we collected a set of 306 *intentional tags* in English including their corresponding translate into Spanish. Perhaps, we could extract more *intentional tags* using vocabulary generation mechanisms, but as a starting point we considered it adequate and we have left this task for a future work.

3.2.2 Users behavior

When we analyzed the user behavior we find that a small number of users (3,08%) used more than 750 different tags, however, 62,8% of users had used less than 10 tags. On the other side, when we look for Marlow's social dimension, we can be compare flikr contacts with CiteULike groups, so, if we isolate data from users who belong to a group (this data were obtained from groups file) we obtain 11.512 users that belong to any group. By extracts the number of labels generated, the number of articles linked, and the number of groups that each user belongs to, we can prove a correlation existence between number of articles linked by a user and the number of tags generated, as well as, the number of articles and the number of groups that user belongs to, in the same way, there is a correlation between the number of tags managed by a user and the number of groups that user belongs to. Table 1 shows correlation Pearson's coefficient for ρ = 0,01.

Also, we can see how users assign tags over time. The next figure shows the number of new tags accumulated by 4 users over time. These users was taken among those who have more than 100 labels and 100 links.

[3]The number of records in the files are updated to January 10, 2011
[4]Source: http://svn.citeulike.org/svn/plugins/HOWTO.txt

Table 1: correlations for ρ = 0,01

	Articles	Tags	Groups
Articles	1	0,662	0,089
Tags	0,662	1	0,144
Groups	0,089	0,144	1

In this figure, we can see two tagging tendencies:

1) Users that periodically generate a great number of new tags in a short period of time (user3 and user4).
2) Users that initially generate a great number of new tags and then decreases their number (user1 and user2).

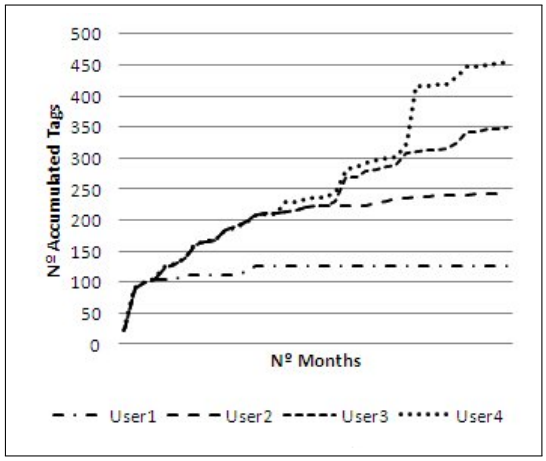


Fig. 1: Nºaccumulated tags/Nºmonths.

The first behavior can be explained by the nature of the site, as it's a site designed for researchers and the academic community, because some users periodically can change their project or their disciplines.

According to Golder [6] and Marlow [7], the second behavior is the most common behavior in social bookmarking tools, you can see a tendency from users to generate a large amount of tags initially and later decrease it as time advances, demonstrating a tendency to reuse the initially generated tags.

3.2.3 Articles

When we analyze the articles, the users behavior is similar to tagging behavior, there are a 52,9% of articles stored by only one user. In the other hand there are a 5,5% of articles stored by 10 or more users, table 2 shows this behavior.

Table 2: Nºusers/article

	NºArticles	Percent
1 user	2.300.649	52,9
2 or 3	1.105.539	25,4
4 or 5	440.256	10,2
6 or 7	175.895	4,0
8 or 9	86.101	2,0
10 or more	240.964	5,5

After this analysis, we consider users behavior in CiteU-Like is similar to users behavior in Delicious or Flikr.

4. Improving the FolkRank algorithm

Once we verified the dataset validity, we'll analyze FolkRank algorithm. FolkRank is a folksonomy practical application, in search and ranking in folksonomy based systems. It's a PageRank based algorithm. Prior to analyzing the algorithm, it is necessary to analyze some previous concepts, necessary for a better understanding of it.

4.1 Folksonomy formal definition

A folksonomy is defined as a tuple $\mathbb{F} = (U, T, R, Y)$ where:

- *U, T, R,* are respectively, sets of users, tags and resources.
- *Y,* is a ternary relation between them, ($Y \subseteq U \times T \times R$) whose elements are called tag assignments.

In this context, $T(u, r)$ is defined like the set of all tags that user *u* has assigned to resource *r* :

$$T(u, r) = \{t \in T | (u, t, r) \in Y\}$$

Then the set of all *post* in the folksonomy is:

$$P = \{u, S, r) | u \in U, r \in R, S = T(u, r), S \neq \emptyset\}$$

In other words, each post contains all tags assigned by one user to one resource.

Then, a folksonomy can be represented as a graph of tags, in which two nodes will be joined by an edge when both tags identify the same article and the weight of that edge is given by the number of users who have used both tags with this article.

Due to this different nature of folksonomies compared to the web graph ("*undirected triadic hyperedges instead of directed binary edges*"), it's necessary adapt PageRank algorithm by convert $\mathbb{F} = (U, T, R, Y)$ into a tripartite graph

$$G_{\mathbb{F}} = (V, E) \quad (1)$$

where V is the set of nodes of the disjoint union between the sets of tags, users and resources ($V = U \dot{\cup} T \dot{\cup} D$), and E is the set of edges representing all co-ocurrences between tags and users, users and resources and tags and resources.

4.2 Recommender systems

Due to FolkRank is a recommendation system, when a user *u* have a resource *r* without any tags assigned to it ($T(u, r) = \emptyset$), it should recommend a set $\tilde{T}(u, r) \in T$ of tags. FolkRank computes a "*topic-specific*" ranking of the elements in a folksonomy, determined by the pair *user/resource* for which the algorithm tries to compute tag recommendation. The rank is computed with the weight spreading computation:

$$\vec{w}_{t+1} \leftarrow dA^T \vec{w}_t + (1 - d)\vec{p} \quad (2)$$

where $\vec{w}$ is a weight vector, d is the damping factor[5] A is the row-stochastic version of the adjacency matrix of the graph $G_{\mathbb{F}}$ and $\vec{p}$ is the random surfer vector, used as preference vector.

The rank is computed through:

$$\vec{w} = \vec{w}^{(1)} - \vec{w}^{(0)} \tag{3}$$

were $\vec{w}^{(0)}$ is the fixed point from equation (2) with $\vec{p} = 1$, and $\vec{w}^{(1)}$ is the same as $\vec{w}^{(0)}$ but $\vec{p}[u] = 1 + |U|$ and $\vec{p}[r] = 1 + |R|$.

Cattuto et al [12] not dismiss intentional tags in their analysis when they used FolkRank algorithm and they include "toread" into most popular tags list. Opposite to this approach we think that this tags should not appear into the set of recommended tags because they do not provide any information about the contents of an article, but in accordance with Kipp and Campbell [10] we think these tags work better as rank modifiers in the way that, when any tag co-occurs with one of these intentional tags, the weight of this edge in the graph must be increased.

At this point, we generated three files with all E co-occurrences of the tripartite graph from equation (1). We have found 512 co-occurrences of our limited set of intentional tags which make us think that this number must grow if we generate a vocabulary for intentional tags.

We propose a FolkRank algorithm modification consisting in changing, by a fuzzy method, the weights of edges between co-occurring tags when an intentional tag co-occurs with both tags. Weightscan be changed by a fixed value or by an age of intentional tag dependent one. The fuzzy changes are made following the initial Kosko's proposals [1][2], and the preliminary tests, according to precision and recall [4] values, shows promising results.

5. Conclusions and future works

In this paper, a proposal for a FolkRank algorithm improvement, based on fuzzy weights modification, has been presented. Future works include to refine the intentional tags set, through vocabulary generation or extraction mechanisms, such as the ones used for information retrieval or search engines tasks (i.e. the ones proposed for our research team in [14] and [9]).

6. Acknoledgements

Partially supported by TIN2010-20395 FIDELIO project, MEC-FEDER, Spain, PEIC09-0196-3018 SCAIWEB-2 excellence project, JCCM, Spain and POII10-0133-3516 PLINIO project, JCCM, Spain.

[5] http://infolab.stanford.edu/~backrub/google.html

References

[1] Kosko, B., Fuzzy Cognitive Maps, Int. Journal of Man-Machine Studies, 1986, 24, pp. 65-75.

[2] Kosko, B., Fuzziness vs. Probability, Int. Journal of General Systems, 1990, 17, pp. 211-240.

[3] Brin, S., Brin, L., "The Anatomy of a Large-Scale Hypertextual Web Search Engine". Computer Networks and ISDN Systems, 30(1–7):107–117, April 1998.

[4] Baeza-Yates, R., Ribeiro-Neto, B., "Modern Information Retrieval". ACM Press, Addison-Wesley, pp.75–81. New York 1999

[5] Vander Wal, T., (2004) Folksonomy Coinage and Definition. [Online]. Available: http://vanderwal.net/folksonomy.html.

[6] Golder, S., and Huberman, B. A., (2005) The Structure of Collaborative Tagging Systems. [Online]. Available: HP Labs technical report. http://www.hpl.hp.com/research/idl/papers/tags/.

[7] Marlow, C., Naaman, M. Boyd, D., et al. *HT06, Tagging Paper, Taxonomy, Flickr, Academic Article, ToRead*. Hypertext 06. http://www.danah.org/papers/Hypertext2006.pdf. 2006.

[8] Hotho, A., Jäschke, R., Schmitz, C., Stumme, G., "Information Retrieval in Folksonomies: Search and Ranking"; Lecture Notes in Computer Science (Springer Berlin Heidelberg), vol. 4011, chap. 31, pp. 411–426. 2006.

[9] Garcés, P., Olivas, J. A., Romero, F. P., "Concept-Matching IR Systems Versus Word-Matching Information Retrieval Systems: Considering Fuzzy Interrelations for Indexing Web Pages"; Journal of the American Society for Information Science and Technology JASIST, 57 (4): 564–576, 2006.

[10] Kipp, M.E., Campbell, D.G., "Patterns and inconsistencies in collaborative tagging systems: an examination of tagging practices"; Proceedings of the 2006 Annual Meeting of the American Society for Information Science and Technology, Austin, Texas,vol. 43, pp. 1–18. 3–8 November 2006. 2006.

[11] Jäschke, R., Marinho, L., Hotho, A, Schmidt-Thieme, L, Stumme, G., "Tag recommendations in social bookmarking systems"; AI Communications, Vol. 21 Issue 4, pp. 231–247, 2008.

[12] Cattuto, C., Benz, D., Hotho, A., Stumme, G., "Semantic Analysis of Tag Similarity Measures in Collaborative Tagging Systems"; In Proceedings of the 3rd Workshop on Ontology Learning and Population (OLP3), Patras, Greece, pp. 39–43, July 2008.

[13] López-Juárez, P., Olivas, J. A., "Estudio del uso del factor tiempo en las folksonomías". In Proceedings of the III Congreso Español de Informática (CEDI 2010). III Simposio sobre Lógica Fuzzy y Soft Computing, LFSC2010 (EUSFLAT), Valencia (Spain), pp. 417–424, 7–10 September 2010.

[14] Serrano-Guerrero, J., Herrera-Viedma, E., Olivas, J. A., Cerezo, A., Romero, F.P., "A google wave-based fuzzy recommender system to disseminate information in University Digital Libraries 2.0"; Information Sciences, Volume 181, Issue 9, 1 May 2011, pp. 1503–1516.

SESSION

FUZZY COGNITIVE MAPS + FUZZY SETS + FUZZY MODELS AND APPLICATIONS

Chair(s)

Dr. Elpiniki I. Papageorgiou
Prof. Jose L. Salmeron

Application of fuzzy cognitive maps using semantic web approaches to model medical knowledge

E.I. Papageorgiou[1], J. De Roo[2], C. Huszka[2] and D. Colaert[2]
[1]Department of Informatics & Computer Technology, Technological Educational Institute of Lamia, Lamia, Greece, epapageorgiou@teilam.gr
[2]Agfa HealthCare NV, Moutstraat 100, 9000 Gent, Belgium, {csaba.huszka, jos.deroo, dirk.colaert}@agfa.com

***Abstract*—*This study presents our investigation into a generic approach to model medical knowledge, using the semantic web framework, to have explicit knowledge with a clear semantic meaning and cognitive maps. Fuzzy Cognitive Maps are proposed to model the medical knowledge extracting by clinical guidelines as they have been proven by literature as powerful knowledge representation and reasoning tools. A general purpose reasoning engine, Eye, with the necessary plug-ins was developed to be able to perform the reasoning on this knowledge model. The concepts and causal relationships among them are transferred in semantic web using the notation3. Our effort is mainly concentrated to present the main aspects of formalizing medical knowledge using cognitive maps and semantic web. The advantage of this approach is to enable the sharing and reuse of knowledge from databases of guidelines and simplify maintenance.*

Keywords*-*fuzzy cognitive maps; knowledge representation; decision support; semantic web; notation3

1. INTRODUCTION

Usually, the decision making process of the treating physician is supported by a large number of medical guidelines produced by medical committees for deciding the relevant therapy to apply to the presented patient [1]. Due to the sheer volume of one such guideline and the rigid structure of such guidelines, it is very difficult for the treating physician to apply and strictly follow the workflow presented by such guidelines. Thus there is a challenge to propose a medical decision support system that dynamically guides the physician through the workflow of guidelines. This workflow is not a static, because the workflow itself is adapting to the changing reality of the patient care [2].

Previous works were mainly based on static modeling of simple rules for medical decision support [3,4]. But the nowadays' medical decision making is focused to more individualized treatment guidelines and the future goal of the research is adjust general guidelines to individual circumstances.

In our approach, in order to keep the main characteristics of uncertainty in focus that is inherent in medical domain, we attempt to model medical knowledge using fuzzy cognitive maps which is a different approach from static structures such as decision trees [5]. FCMs have been proved from the literature as efficient medical decision-making and support techniques [6-9]. So currently, we focus on formalization of medical knowledge using dynamic influence graphs, such as FCMs, which create paths, based on the current and constantly changing clinical information of the patient, the environment and the guidelines.

The undertaken work is focused on the establishment of the generic knowledge representation model using Fuzzy Cognitive Maps, implemented in the semantic web framework. Notation3 [10] was selected as an open and semantic web language to implement the fuzzy cognitive maps. A general purpose reasoning engine, Eye [11], and the necessary plug-ins to be able to perform the reasoning on these knowledge models were also developed and presented. The suggested approach is implemented to the Urinary Tract Infection problem caused by Escherichia coli and the preliminary trials show its effectiveness in medical decision support.

2. MODELING MEDICAL KNOWLEDGE

In order to model medical knowledge it is essential to deal with the causal feature of the knowledge elements. It is well known that medical information is abundant and ever-growing, yet it hardly represents a complete set of knowledge, rather a limited set thereof. Medical knowledge can be represented in several ways. Perhaps the most commonly used is the clinical pathway or clinical guideline [12,13]. It is quite easy and handy to use by the clinician as it provides a simple decision tree algorithm that could be followed in the diagnostic or therapeutic path to aid conclusions. However, many guidelines are not so easy to use in daily practice due to the complexity of the clinical data [1,2]. Thus, there is a need to create a decision support system (DSS) that aims to recognise the appropriate guidelines in front of a medical problem.

2.1 Medical Models

Modelling medical knowledge starts with a standard procedure, by defining the problem to be modelled (as a domain) and identifying all variables that represent this domain. Our model follows a goal-drive, need-oriented approach meaning that each model will serve a certain purpose and is custom-made. We identified two basic models to be used in the scope of this study: diagnostic model and therapeutic model.

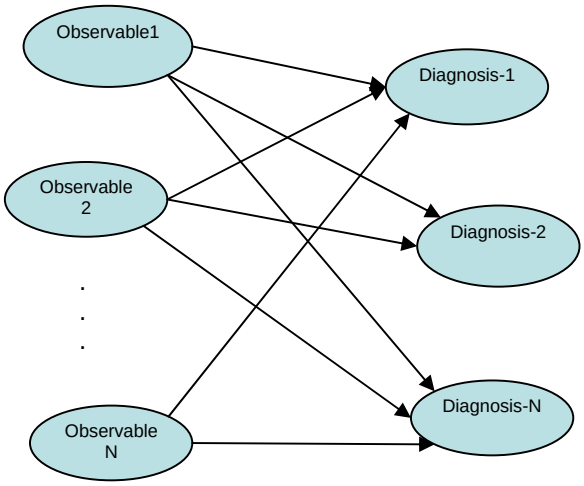


Figure 1. Generic structure of a diagnostic model.

2.2 Diagnostic Model

This model represents diagnostic decision patterns. Its observables describe mostly sign/symptoms and/or examination procedures. The aim of the model is to represent a diagnostic process within a domain (which is usually a medical condition or disease) giving each and every step of the diagnostic act a belief.

2.3 Therapeutic Model

This model represents therapeutic patterns. It is designed to aid the physician in selecting the most appropriate therapy for his patient. In this model observables are defined around the domain of a therapeutic process. Since the same therapy might have different scope and features when applied to different medical conditions and diseases, it is necessary to properly identify the model's scope. This "combinational representation" of domain is necessary in order to ensure that the scope of medical therapy is properly covered. The variables within this domain will be observables from the scope of a therapy (e.g. indications, contra- indications, relative indications, relative contra-indications etc.).

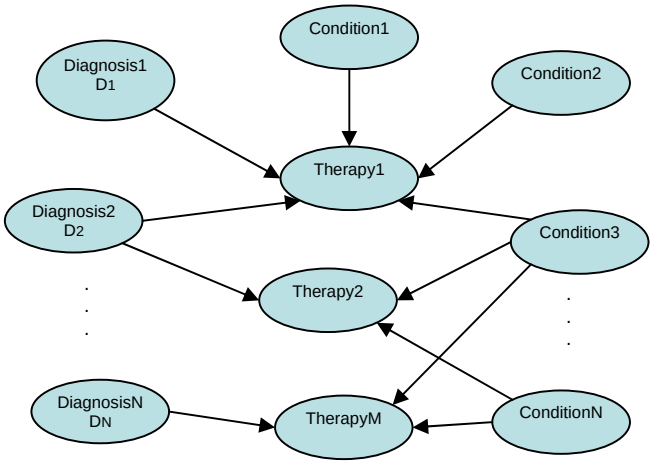


Figure 2. Generic structure of a therapeutic model.

Quantifying the relationship between these observables and the therapy is based on clinical trials, clinical studies and/or expert opinion. In next section the quantification of these relationships is accomplished using the FCM methodology and then the semantic web language N3 is used to implement them.

3. Fuzzy Cognitive Maps and Semantic Web Tools

3.1 Fuzzy Cognitive Map

FCMs constitute an extension of cognitive maps, inheriting the main aspects of fuzzy logic and neural networks. They are ideal causal knowledge acquiring tools with fuzzy signed graphs which can be presented as an associative single layer neural network [5]. They describe particular domains using nodes (variables, states, inputs, outputs) and signed fuzzy relationships between them. The fuzzy part allows us to have degrees of causality, represented as links between the nodes of these diagrams, also known as concepts. This structure establishes the forward and backward propagation of causality, admitting the knowledge base to increase when concepts and links between them are increased [9,14].

The fuzzy relations between concepts can be used to compute the strength of impact of these concepts. Similarly to the FCM, this method will also require a through identification of variables-concepts and their connections, however the relationship is described with numeric value in the range of -1;1 instead of a non-negative real number.

3.2 Semantic Web

The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries [15]. It is based on the Resource Description Framework. The framework enables reasoning as there is now a clear distinction between a 'world' and an explicit and formal semantic theory about that 'world'.

Reasoning

The chief utility of a formal semantic theory (as mentioned in "Semantic Web") is to provide a technical way to determine when inference processes are valid, i.e. when they preserve truth. This provides the maximal freedom for reasoning engines while preserving a globally coherent notion of meaning.

3.3 Eye reasoner

Eye stands for "Euler yap engine" and it is a further incremental development of Euler which is an inference engine supporting logic-based proofs. Eye is a backward-forward-backward chaining reasoner design enhanced with Euler path detection.

The backward-forward-backward chaining is realized via an underlying Prolog backward chaining, a forward meta-level reasoning and a backward proof construction. The Euler path detection is roughly "don't step in your own steps" to avoid vicious circles so to speak and in that respect there is a similarity with what Leonhard Euler discovered in 1736 for the Königsberg Bridge Problem [16][16].

Eye is the latest implementation of the Euler proof engine and is released as open source on http://eulersharp.sourceforge.net. A detailed log of the changes can be found at http://eulersharp.sourceforge.net/DONE. More architectural/technical background for Eye Reasoner is given in [11].

3.4 Notation3 (N3)

It is a logical platform that uses semantic web notation 3 as RDF syntax and expands the vocabulary of predicates. The goal of logic N3 is possible to integrate an RDF data model, logical rules and provide integrated functions that allow access and reason about data [10]. If you want to compare RDF to N3, you can say that in RDF, information is simply a collection of statements, each with a subject, verb and object - and nothing else where in N3, you can write an RDF triple just like that.

```
:patient :has :fever
```

In a rule, premise is the subject and the conclusion is the complement (object). The sign "=>" is a common predicate expression log-rule namely **implies.**

4. FCM IMPLEMENTATION IN N3

The FCM formalization and reasoning was also implemented in N3, using plug-ins and built-ins of the inference engine EYE (section III.C).

4.1 Introducing the FCM built-ins

Medical models are written in a form of T-rules which simplifies the modelling technique to expressing relationships with True or False structures and attaching believes to them. Furthermore the rules expressing medical knowledge make use of built-ins of the Eye reasoner. In the case of FCM modelling the following Eye predicates are used with the Euler built-ins [17,18]:

`fl:mu` is an Eye predicate to express fuzzy set membership, e.g. (:x :C) fl:mu 0.8 says that :x is a :C to a fuzzy membership degree of 0.8.

`fl:sigma` is an Eye predicate to express fuzzy subsethood e.g. (:C :D) fl:sigma 0.9 says that :C is a rdfs:subClassOf :D to a degree of 0.9.

`fl:pi` is an Euler built-in rdf:Property to express the reasoning process of FCM. fl:pi is a built-in supplied via plug-in [18].

In FCM rules and reasoning [17], the main aspects of programming FCM approach in N3 are presented.

4.2 Representing Knowledge using cognitive maps

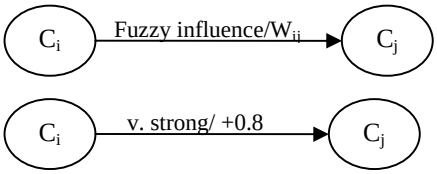
The formalized medical knowledge in the context of FCMs is represented with ontological concepts using N3 and logic-rules format. The construction of an FCM for the modeling of a medical decision making task is implemented in two steps:

The *concepts* (N) have to be provided by medical experts and/or guidelines that sufficiently describe the decision making task, including the input and the output knowledge. Each concept is modeled as a *variable* C_i, i=1,2,…N that can take fuzzy or discrete values according to the problem data. The fuzzy values express the degree to which the concepts occur.

The *connections* between the concepts and their *strenghts* should be assigned also by medical guidelines and/or physicians' knowledge using if-then rules, that infer a fuzzy linguistic variable from a determined set, namely T{influence}, which associates the relationship between the two concepts and determines the grade of causality between the two concepts. These fuzzy sets express the degree to which a concept Cj influences another concept Ci, i=1,2,…N, j=1,2,…N. Its term set T{influence} is suggested to comprise twelve variables. Using twelve linguistic variables, an expert can describe in detail the influence of one concept on another and can discern between different degrees of influence. The twelve variables used here are: *T(influence) = {negatively very strong, negatively strong, negatively medium, negatively weak, negatively very weak, zero, positively very weak, positively weak, positively medium, positively strong, positively very strong, positively very very strong}*.

Generic examples on how medical models are represented using FCMs and then implemented in N3 are given.

Let's consider an FCM with two concepts, C_i and C_j with a fuzzy influence between these concepts. The fuzzy influence can be positive or negative and could be determined from the fuzzy set *T(influence)*.



In semantic web, the above FCM is represented:

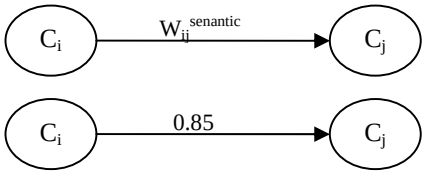


Figure 3. Example of FCM causal relationships implemented in semantic web.

And implemented in N3 as:

```
(:Ci :Cj) fl:sigma Wij semantic.
```

Or

```
(:Ci :Cj) fl:sigma 0.85.
```

The concepts Ci and Cj are categorized in four categories according to the variable they represent; *Observables* - O_i (usually represent symptoms/signs and/or lab exams), *Conditions*-C_i which are related observables defined around the

domain of a therapeutic process and might be for example indications, contraindications, relative indications, relative contraindications, *Diagnosis-Di* (represent the type of diagnosis) and *Therapy-Ti* (represent the suggested therapy for specific patient case).

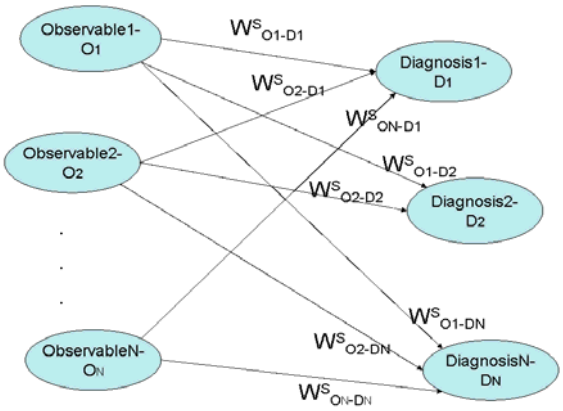


Figure 4. Example of a FCM-based diagnostic model in semanctic web.

Lets consider that the concept Ci represents an *Observable* and concept Cj represents a *Therapy*.

This can be represented in N3 using the following muster:

```
(:Observable :Therapy) fl:sigma 0.XX.
```

Where `fl:sigma` expresses a "subsethood" between the 2 variables (therapy and observable) which can be interpreted as how much the observable (which is defined around the domain of a therapeutic process and might be for example indications, contraindications, relative indications, relative contraindications etc.) is contained in the set of therapy.

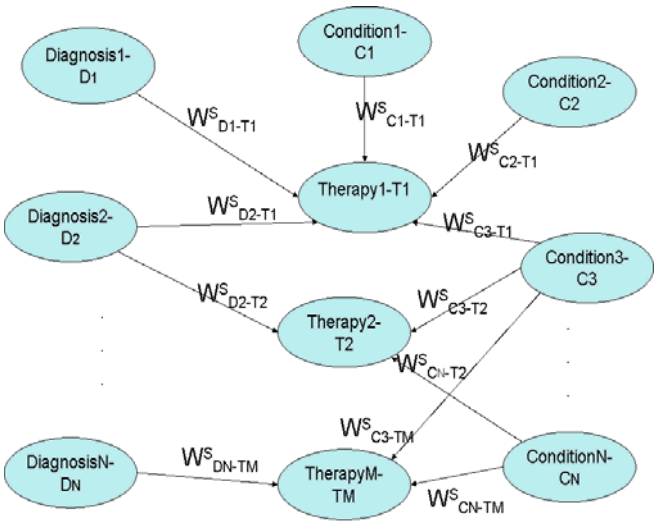


Figure 5. Example of a FCM-based therapeutic model in semantic web.

Quantifying the relationship between these observables and the therapy is based on clinical guidelines, studies and/or physician opinion. As it is noticed this requires a fuzzy value, which can be derived according to the methods mentioned above.

The FCM concepts are included in an ontology repository [20] established for the project needs, namely DCO. DCO is updated continuously in order to contain all relevant clinical concepts [19]. Thus, we have

```
(doc:Observable dco:Therapy) fl:sigma 0.XX.
```

We worked with the Debugit project ontologists to enrich the DCO ontology that can be used with our rules.

Thus the FCM diagnostic model in Fig. 1, is translated to Fig. 3, depicting the main aspects of semantic web approaches. The same happens with Fig. 2, which is translated in semantic web, through the Fig. 4.

The model in Fig. 5 is implemented in N3 in what follows:

```
(:D1 :Therapy1) fl:sigma W^S_D1-T1.
(:D2 :Therapy1) fl:sigma W^S_D2-T1.
(:D2 :Therapy2) fl:sigma W^S_D2-T2.
...
(:DN :TherapyN) fl:sigma W^S_DN-TN.
(:Condition1 :Therapy1) fl:sigma W^S_C1-T1.
...
(:ConditionN :TherapyN) fl:sigma W^S_CN-TN.
```

To implement the initial values of concepts *Ci*, namely *Ai*, in N3, we introduced the Eye predicate `fl:mu`, which express fuzzy set membership,

e.g. `(:patient001 :Fever) fl:mu 0.YY` says that `:patient001` has `:Fever` to a fuzzy membership degree of 0.YY.

Lets consider that the patient has "high fever". The observable "Fever" takes five fuzzy values from the corresponding fuzzy sets {"hypothermia" (34-36), "normal" (36-38.4), "low" (38.5,38.9), "moderate" (38.9,39.5), "high" (39.5-40.9), "hyperpyrexia" (>41)}. These fuzzy values are transferred to numerical values in the Table I.

TABLE I. EXAMPLE OF FUZZY SETS

Fuzzy membership functions for Fever (means Temperature(^{0}C))	***Fuzzy regions (^{0}C)***	***Fuzzy membership degree***
hypothermia	(34-36)	0
normal	(36-38.4)	0.1
low	(38.5-38.9)	0.3
moderate	(38.9-39.5)	0.5
high	(39.5-40.9)	0.8
hyperpyrexia	(>41)	1

Thus for high Fever, the initial value of concept Fever is 0.75 and in FCMs is represented as $Ai(0)=0.8$. In N3, the initial value of observable Fever is implemented as:

```
(:patient001 :Fever) fl:mu 0.8.
```

4.3 FCM reasoning in N3

This value *Ai* of each concept C_i is updated at next step k, following the FCM reasoning process [9] that is described at next equation:

$$A_i^{(k+1)} = f\left((2A_i^{(k)} - 1) + \sum_{\substack{j \neq i \\ j=1}}^{N} (2A_j^{(k)} - 1) \cdot W_{ji}\right) \quad (3)$$

The implementation of the FCM reasoning in N3 is accomplished with the fcm-plugin.yap [18]. `fl:pi` is an euler built-in `rdf:Property` to express the inference algorithm of FCM, referring to a number of iterations as it is expressed in eq. (3).

4.4 Modeling influences in N3

In N3, the positive and negative influences expressed by FCM weights should be translated in the range 0 to 1, keeping their meaning (see Table II). Each one of the FCM weights (concepts' influences) presents a degree of belief from 0 to 1. Thus, using the Eye predicate `fl:sigma` the weight is formalized in N3.

4.5 Formalization of rules in N3

Using N3, the rules and conditions for FCM are formalized in N3 and a number of concrete examples are given at follows:

```
IF the patient has Allergy to Nitrofurantoin
THEN Nitrofurantoin is inadequate for UTI caused by E.coli infection
```

Translated in N3 using T-rules like:

```
{(?P :AllergyToNitrofurantoin) fl:mu 1} => {(:UrinaryTractInfection :Nitrofurantoin) fl:sigma 0}.

{(?P :AllergyToNitrofurantoin) fl:mu 1} => {(:EscherichiaColiInfection :Nitrofurantoin) fl:sigma 0}.
```

TABLE II. NUMERICAL WEIGHTS FOR SEMANTIC WEB

Fuzzy membership functions (triangular)	*Fuzzy regions*	*Deffuzzified value (weight)*	*Semantic weights*
Negative Very very strong influence	[-1, -0.9)	-1	0
Negative Very strong influence	(-0.9 -0.7)	-0.8	0.01
Negative. strong influence	(-0.8 -0.6)	-0.7	0.05
Negative medium influence	(-0.7 -0.4)	-0.55	0.1
Negative weak influence	(-0.4 -0.2)	-0.3	0.15
Negative Very weak influence	(-0.2 0)	-0.1	0.2
Positive Very weak influence	(0 0.2)	0.1	0.3
Positive weak influence	(0.2 0.4)	0.3	0.4
Positive medium influence	(0.4 0.7)	0.55	0.55
Positive strong influence	(0.6 0.8)	0.7	0.75
Positive Very strong influence	(0.7 0.9)	0.8	0.9
Positive Very very strong influence	(0.9 1]	1	1

Using again the same explanation way, we separate the code to three parts as shown in Fig.6.

```
Subject: A={(?P :AllergyToNitrofurantoin) fl:mu 1}
Predicate: => implies
Object: B= {(:UrinaryTractInfection
:Nitrofurantoin) fl:sigma 0}
```

Figure 6. Example of T-rule for N3.

Based on the *if-then* rule syntax, the patient is presented as ?P ; this means that the rule is activated and have to be satisfied for all the patient cases, i.e. `patient001, patient002` e.t.c.

The second part of the rule contains the **"=> `imply`"** symbol that leads to the consequences; what is going to be happen if `allergy to Nitrofurantoin` is detected in the patient's input data (patient state)?

It is expressed by the Object, triplet B `({(:UrinaryTractInfection :Nitrofurantoin) fl:sigma 0})`. In this statement we set the value of the therapy impact to the disease, equal to 0. Due to allergy, the treatment with this antibiotic is extremely harmful for the patient. Another example on formalization of if-then rules in N3 is presented in Fig. 7.

Example Rule-1: `If the patient is pregnant Then TMP/SMX is inadequate`
Formalization in N3:

```
{(?P :PregnancyBefore28Wk) fl:mu 1. (?P :PregnancyBetween28-34Wk) fl:mu 1. (?P :PregnancyAfter34Wk) fl:mu 1} => {(:EscherichiaColiInfection :TMP_SMX) fl:sigma 0}.

{(?P :PregnancyBefore28Wk) fl:mu 1.(?P :PregnancyBetween28-34Wk) fl:mu 1.(?P :PregnancyAfter34Wk) fl:mu 1} => {(:UrinaryTractInfection :TMP_SMX) fl:sigma 0}.
```

Figure 7. N3 formalization of example Rule-1.

This code follows the principles of the above example with `AllergyToNitrofurantoin`, with the difference that the condition of pregnancy is composed of three concepts: `PregnancyBefore28Wk`, `PregnancyBetween28-34Wk` and `PregnancyAfter34Wk`. The union of these concepts with the way of expressing 1 condition was made by using the N3 syntax symbol **dot** "."

5. MEDICAL EXAMPLE AND RESULTS

The uncomplicated Urinary Tract Infection (UTI) therapy caused by Escherichia coli (e. coli) decision making problem was considered to show the implementation of the proposed approach. UTI can be classified as uncomplicated (patients with urinary tracts that are normal from both structural and functional perspective) and complicated [21,22]. The UTIs caused mainly by Escherichia coli are most common in young, sexually active women (due to shorter urethra) as well as in hospitalized patients [22].

Treatment of uncomplicated UTIs ranges from conservative therapy (abundant fluid intake) to antibiotic therapy. There are several medications available for oral therapy of such as trimethoprim (TMP), cephalosporine, nitrofurantoin, fluoroquinolone. Combined form of Trimethoprim with Sulfametoxazol (TMP/SMX) is one widely used antibiotic for UTIs and suggested as *first line* therapy. Fluoroquinolones (2nd and 3rd generation drugs such as levofloxacin or ciprofloxacin) should be preserved (as backup) for patients with allergies or with underlying diseases that predispose to serious condition as well as for patients where first line drugs fail. Nitrofurantoin and fosfomycin may become more useful as resistance to TMP-SMX and trimethoprim increase [22].

The formal therapy model using sub-FCMs for each one of the empirical therapies for the uUTI treatment was developed. The therapy model was designed with the contribution of medical guidelines/treatment suggestions in [22] and reasoned following the methodology described in Section IV.C (based on Eye reasoning engine presented in section III.C).

Due to aim of this study to present a generic approach on formalizing medical knowledge, and in order to avoid the complexity to present the whole development of the formal therapy model, we focused to demonstrate the modelling process for the treatment of UTI with TMP/SMX (shown in Figure 8) by using sub-FCM TMP/SMX therapy model as example.

The relationships (weight values) between the therapy conditions and treatment TMP/SMX were formalized in N3 and depicted in Fig. 9. The results presented here follow the steps of the methodology previously described and the inference mechanism of FCMs (link to Eye reasoning engine of eulersharp/plugin). To show how the inference works, lets consider a patient with age below 18 years old suffers from UTI caused by e. coli. The initial state of the patient is formalized in N3 at Fig. 10.

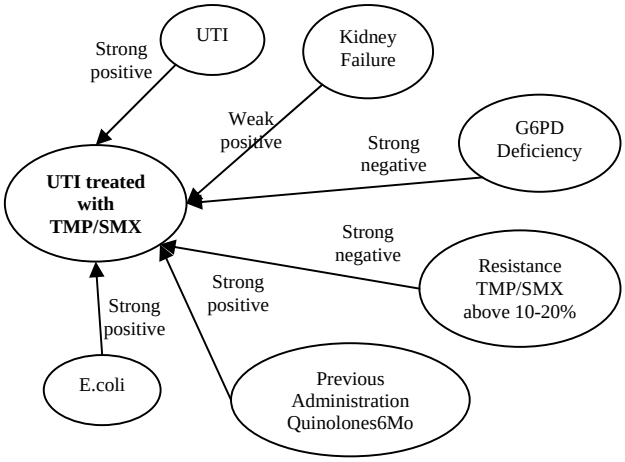


Figure 8. Sub-FCM model for TMP/SMX therapy.

The query in N3 is established as follows:

```
@prefix fl: <http://eulersharp.sourceforge.net/2003/03swap/fl-rules#>.
{?X fl:pi ?Y} => {?X fl:pi ?Y}.
########
```

```
# TMP_SMX Therapy Model

@prefix math:
<http://www.w3.org/2000/10/swap/math#>.
@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix fl:
<http://eulersharp.sourceforge.net/2003/03swap/fl-
rules#>.
@prefix : <fcm#>.

(:G6PDDeficiency :TMP_SMX) fl:sigma 0.
(:KidneyFailure :TMP_SMX) fl:sigma 0.4.
(:AntibiogramAvailab :TMP_SMX) fl:sigma 0.9.
(:PreviousAdministrationOfQuinolones6Mo :TMP_SMX)
fl:sigma 0.9.
(:PregnancyBefore28Wk :TMP_SMX) fl:sigma 0.
(:PregnancyBetween28-34Wk :TMP_SMX) fl:sigma 0.
(:PregnancyAfter34Wk :TMP_SMX) fl:sigma 0.
(:AllergyToTMP_SMX :TMP_SMX) fl:sigma 0.
(:UrinaryTractInfection :TMP_SMX) fl:sigma 1.
(:EscherichiaColiInfection :TMP_SMX) fl:sigma 1.
(:resistance_TMP_SMXabove10-20Pct :TMP_SMX) fl:sigma
0.
{(?P :AllergyToTMP_SMX) fl:mu 1} =>
{(:UrinaryTractInfection :TMP_SMX) fl:sigma 0}.
{(?P :AllergyToTMP_SMX) fl:mu 1} =>
{(:EscherichiaColiInfection :TMP_SMX) fl:sigma 0}.
```

Figure 9. TMP/SMX therapy model formalized in N3.

Then, using the Eye plugin and reasoning engine [17,18], the tool gives the answer for this patient case. We present an overview of the RDF file of the results given by the inference engine supplied via plug-in in Fig. 11.

It is concluded that the patient001 is very likely to receive Amoxicillin as suggested treatment therapy. Amoxicillin is presented as the appropriate therapy for this patient case. Except Amoxicillin as the first line therapy, Nitrofurantoin, Ciprofloxacine, Lomefloxacine, Norfloxacine, Ofloxacine, and Fosfomycin could be also suggested as second-line therapies according to other patient's conditions.

The proposed therapy models give an order of possible antibiotic treatments, instead of one, as it really happens in clinical practice. It suggests first line and second line therapies. This work presents our preliminary results performed from the proposed approach. Of course, more clinical trials and patient cases are needed to be experimentally analyzed to prove the usefulness of the proposed approach. Future work is directed towards this and new clinical applications will be investigated.

```
@prefix fl:
<http://eulersharp.sourceforge.net/2003/03swap/fl-
rules#>.
@prefix : <fcm#>.
(:patient001 dco:UrinaryTractInfection) fl:mu 1.
(:patient001 dco:EscherichiaColiInfection) fl:mu 1.
(:patient001 dco:AgeBelow18) fl:mu 1.
```

Figure 10. Example of initial patient state formalized in N3.

6. CONCLUSIONS

This study presents a generic approach on how the medical knowledge could be formalized using cognitive maps and semantic web. Especially the N3, was proposed to implement

```
   #gives the answer for this patient001
(:patient001  :Nitrofurantoin) fl:pi
0.832018385133924.
(:patient001  :Ciprofloxacine) fl:pi
0.832018385133924.
(:patient001  :Lomefloxacine) fl:pi
0.832018385133924.
(:patient001  :Norfloxacine) fl:pi 0.832018385133924.
(:patient001  :Ofloxacine) fl:pi 0.832018385133924.
(:patient001  :Fosfomycine) fl:pi 0.832018385133924.
(:patient001  :TMP_SMX) fl:pi 0.310025518872388.
(:patient001  :Amoxicilline) fl:pi 0.952574126822433.
```

Figure 11. Overview of the RDF file with results produced by the proposed approach.

the FCM variables, namely concepts and relationships among them, namely weights. A reasoning engine, Eye, with the necessary plug-ins was developed to be able to perform the reasoning on this knowledge model. The proposed methodology was applied in the decision making problem of uncomplicated UTI caused by e.coli therapy. A number of medical guidelines for the therapy of uUTI were formalized into elementary FCMs that constitute the formal knowledge models. Based on FCM formalism and semantic web approach a preliminary decision support tool for UTI treatment is presented by proposing the appropriate antibiotic(s) for each patient case.

ACKNOWLEDGMENT

The DebugIT project (http://www.debugit.eu/) is receiving funding from the European Community's Seventh Framework Programme under grant agreement n° FP7–217139, which is gratefully acknowledged. The information in this document reflects solely the views of the authors and no guarantee or warranty is given that it is fit for any particular purpose.

REFERENCES

[1] E.S. Berner, Clinical Decision Support Systems: Theory and Practice, Springer, 2007

[2] W.M. Tierney, J.M. Overhage, B.Y. Takesue, et al. "Computerizing guidelines to improve care and patient outcomes: the example of heart failure", *J. Am. Med. Inform. Assoc.*, vol. 2(5), 1995, pp. 316-322.

[3] Johnston ME, Langton KB, Haynes RB, Mathieu A. Effects of computer-based clinical decision support systems on clinician performance and patient outcome. Ann Intern Med 1994; 120: 135-142

[4] C.A.M. Schurink, P.J.F. Lucas, I.M. Hoepelman, and M.J.M. Bonten, "Computer-assisted decision support for the diagnosis and treatment of infectious diseases in intensive care units", The Lancet of infectious diseases, Volume 5, Issue 5, May 2005, Pages 305-312.

[5] B. Kosko, "Fuzzy Cognitive Maps", *International Journal of Man-Machine Studies*, vol. 24, 1986, pp. 65-75.

[6] R.I. John, P.R. Innocent, "Modeling Uncertainty in Clinical Diagnosis using Fuzzy Logic," *IEEE Transactions on Systems, Man, and Cybernetics*, Part B, vol. 35, no. 6, 2005.

[7] E.I. Papageorgiou, N.I. Papandrianos, G. Karagianni, G.C. Kyriazopoulos, D. Sfyras, "A Fuzzy Cognitive Map based Tool for Prediction of Infectious Diseases," in Proc. *IEEE International Conference on Fuzzy Systems* (FUZZ), 2009, pp. 2094-2099.

[8] C. Stylios, V. Georgopoulo Fuzzy Cognitive Maps Structure for Medical Decision Support Systems, in: Studies in Fuzziness and Soft Computing, vol. 218, pp. 151-174, Springer, 2008.

[9] E.I. Papageorgiou, "A New Methodology for Decisions in Medical Informatics using Fuzzy Cognitive Maps based on Fuzzy Rule-Extraction Techniques", *Applied Soft Computing*, vol. 11, pp. 500-513, 2011.

[10] Tim Berners-Lee, Dan Connolly, Notation3 (N3): A readable RDF syntax, http://www.w3.org/TeamSubmission/n3/

[11] J. De Roo, Eye, available in **http://eulersharp.sourceforge.net/2003/03swap/eye-2009.txt**

[12] D.K. Owens, R.D. Shacher, R.F. Nease, "Representation and analysis of medical decision problems with influence diagrams", Med. Decis. Making 17 (1997) 251—262.

[13] R.F. Nease, D.K. Owens, Use of influence diagrams to structure medical decisions, Med. Decis. Making 17 (1997) 263—275.

[14] Salmeron, J.L. Supporting decision makers with fuzzy cognitive maps: These extensions of cognitive maps can process uncertainty and hence improve decision making in R&D applications. 2009 Research Technology Management 52 (3), pp. 53-59.

[15] Tim Berners-Lee, The Semantic Web as a language of logic, available in **http://www.w3.org/DesignIssues/Logic.html**

[16] Weisstein, Eric W, Königsberg Bridge Problem. From MathWorld --A Wolfram Web Resource. **http://mathworld.wolfram.com/KoenigsbergBridgeProblem.html**

[17] http://eulersharp.sourceforge.net/2003/03swap/fl-rules.n3

[18] http://eulersharp.sourceforge.net/2006/02swap/fcm-plugin.yap

[19] http://www.imbi.uni-freiburg.de/ontology/dco/dco

[20] OWL web language, available in **http://www.w3.org/TR/2004/REC-owl-features-20040210/**

[21] L. E. Nicolle, "Urinary tract infection: traditional pharmacologic therapies", *Dis Mon*, vol. 49, 2003, pp. 111-28.

[22] J. W. Warren, E. Abrutyn, J. Richard Hebel, J. R. Johnson, A. J. Schaeffer, and W. E. Stamm, "Guidelines for Antimicrobial Treatment of Uncomplicated Acute Bacterial Cystitis and Acute Pyelonephritis in Women", Clinical Infectious Diseases vol. 29, 1999, pp. 745–58.

Creating a Suicide Note Analysis Model Using Fuzzy Cognitive Maps

Ethan White[1] and Lawrence J. Mazlack[2]
Applied Computational Intelligence Laboratory
University of Cincinnati, Cincinnati, Ohio USA
[1]whitee4@mail.uc.edu
[2]mazlack@uc.edu

Abstract – Suicide has become a significantly prominent issue because high frequencies of occurrences make it one of the top three causes of death for young people in the United States. To prevent suicide attempts, it is important to identify suicidal tendencies in the behavior, speech, or writing of an individual as early as possible. This paper demonstrates one technique of analyzing written material to determine whether or not a person is in a suicidal state. Word frequencies were examined from both suicide and non-suicide notes and translated into inputs to a fuzzy cognitive map (FCM). For the datasets examined, each set was correctly identified as suicidal or non-suicidal.

Keywords: fuzzy cognitive maps (FCM); causality; suicide; language; natural language processing

1 Introduction

Computationally recognizing causality is a difficult task that has great potential to be one of the most useful predictive tools for analyzing real world events. One utilitarian outcome that causality provides is the prediction of human behavioral patterns either in broad domains such as nations and religious groups, or in groups of individuals. One such group of individuals that can be studied is those people who commit suicide. Suicide is one of the top three causes of death in the United States for 15-34 year olds [19], making it a pertinent topic for study. One of the ways to study suicide is analysis of the notes that were left behind by the persons who committed suicide [3]. Although a quantitative description of causality in suicide notes is difficult, the analysis can be represented qualitatively. The purpose of this work is to use fuzzy cognitive maps to discover and isolate root causal relationships based on words in suicide notes from people who take their own lives.

The long term goal of this work is to discover patterns within written material in such a way as to determine causal relationships in human behavior. The focus is on patterns in the frequency of words as opposed to grammatical structure. The purpose of this present work, the first step toward the long term goal, is to analyze a specific human behavioral pattern, i.e. suicide, in the form of suicide notes in such a way as to contrast them with non-suicide notes. The central hypothesis is that suicidal behavioral patterns can be extracted from word frequencies in written material, and that these patterns can be represented using fuzzy cognitive maps.

To test the central hypothesis and accomplish the overall purpose of this research, three specific objectives are pursued. The first objective is to discover and extract patterns in written material in order to produce an initial fuzzy cognitive map that describes the causality. The first step toward this objective is the analysis of suicide notes according to the working hypothesis that the causal patterns can be discovered by finding word frequency patterns. The second step of objective one is an analysis of non-suicide notes based on the same working hypothesis. Once this analysis is accomplished, the frequency patterns of the data will be used to produce an initial fuzzy cognitive map for analysis in objective two.

The second objective is to test the fuzzy cognitive map on the original datasets and make adjustments until it can successfully identify each set as suicidal or non-suicidal. Once the first objective has been accomplished and the patterns discovered are converted to a fuzzy cognitive map, then testing must be performed in order to ensure that the map will be able to distinguish the difference between the suicide notes and the non-suicide notes that were originally tested. Once the map has been modified to a point where the results are acceptably reliable then objective three will be performed.

Objective three is to perform rigorous testing on the fuzzy cognitive map based on different material. Once the cognitive map is able to distinguish between the original data sets used to build it, the map must be able to find the patterns in different written sources to make sure that it can work on a variety of writing. If satisfactory results have not been attained, then the new data must be factored into the fuzzy cognitive map until the results are reliably accurate. Then objective three must be performed again using a different source of data.

2 Background

2.1 Suicide research

Much of the research in studies of suicide occurs in two major areas: clinical studies and analysis of suicide notes.

There are generally two types of clinical studies. The first type is a statistical analysis, and it usually involves analyzing suicides based on age and gender as well as other demographics such as social and educational status [5]. This method has been traditionally a primary means of analysis in researching suicide [6]. For example, some studies have shown that nearly one-third of all suicides are found in the elderly over 60 years of age [1]. By using this technique, it is possible to discover what kinds of people are more likely to commit suicide. This approach allows researchers and suicide prevention organizations to identify particular trends in the demographics where they can focus their work in discovering ways to prevent such suicides.

The second type of clinical study involves a psychological analysis of individual patients who are displaying suicidal thoughts and actions, discovering reasons for committing suicide on an emotional and psychological level. This method is done by studying people with suicidal thoughts in order to correlate emotional, mental, or physical states with suicide. An example would be to study the emotional state of people who are losing eye sight, who especially have a fear of losing sight, and how it correlates with suicidal thoughts [7]. If the thought processes of people who are undergoing blindness can be thoroughly understood, than it would be possible to anticipate and treat psychopathological symptoms before they lead to suicide [7].

The analysis of suicide notes has been primarily a study into the content of the notes in an attempt to discover the state of the person's thinking and thereby understand what circumstances and emotional states were driving the individual to commit suicide. These circumstances can be divided up into situations, relationships, emotional state, cognitive state, and demographic variables [3]. These different aspects are then analyzed to determine causes and motivations for suicide similar to the clinical studies approach.

2.2 Causality

Causality is simply a description of what causes a particular event or group of events in order to understand why something has happened [10]. However, it is not a statistical relation since it contains both a direction, i.e. one event causes another event and not the other way around, and it also has a particular order in a string of events that, if changed, would alter the meaning of the events [10]. Causality extracted from outside information would help intelligent systems model their environment for decision making [11]. Also, causal relationships are more stable in describing the world than probabilistic models [12]. However, constructing causal relationships is difficult because often times it is difficult to determine whether a particular event is necessary for the causal relationship or is merely coincidental [13]. Two methods that are used for modeling causality are text information extraction and graphs.

One of the challenging aspects of analyzing language is in the fact that language is highly complicated and often ambiguous making it difficult for computers to understand [14]. However, extracting causal knowledge from text is important because the internet has provided a huge amount of text that is machine-readable. These documents have a great deal of information in them that can be used to create causal models, e.g., analyzing newspaper articles in order to determine causal economic trends [15]. Another use is in extracting causal patterns from text documents in order for computers to answer questions by analyzing lexical and semantic constructions within the sentences [16].

Graph structures can be used for both modeling outside events, such as Mideast relations [9], as well as describing causal relations in textual data [17]. The reason why these networks are so powerful is that the graphs are made up of simple elements but create complex models that can be used to describe complex relationships [18]. One type of graph structure that can capture causality is a fuzzy cognitive map.

2.3 Fuzzy Cognitive Maps

Fuzzy cognitive maps are a fairly recent development in the attempt to capture causality in a way that can be useful for studying the real world. These maps are directed weighted graphs that describe relationships between events. The nodes and edges together capture domain information allowing the fuzzy cognitive map to depict uncertain causal reasoning [8]. The individual nodes represent particular events and the directed weighted edges represent what kind of an influence each node has on the other nodes connected to it. For example, if node one has a positive influence on node two and if the event of node one increases, then the event of node two will also increase; if they are connected with a negative edge and if node one increases, then node two will decrease [9]. Each node in the graph is a fuzzy set that goes from 0% to 100% [9]. Also, each edge can be a set of concepts such as "a little" up to "a lot" or a range of numbers such as 0% to 100% [9] [2].

3 Creating the Initial FCM

3.1 Extracting patterns in general categories from written material

The first step in accomplishing the first objective and developing the initial fuzzy cognitive map was to analyze the written data of both suicide notes and non-suicide notes. The group of suicide notes that was studied consisted of notes written by those who successfully committed suicide. The non-suicide notes consisted of three sets that were approximately the same size as the number of words used in the group of suicide notes.

All three sets are taken from informal sources, i.e., each source represented a casual form of communication as opposed to magazine articles, professional journals, and other such written works. The first set is a collection of various product reviews extracted from www.Amazon.com. This sample set was taken from a number of different products over a range of different ratings from the highest rating of five stars to the lowest rating of one star. The second set is a collection of notes from a private blog at archbishop-cranmer.blogspot.com. This set is different from the Amazon.com data because it represents an individual instead of a group of people. The final set comes from a political website, www.biggovernment.com. This set contains more specific topics than are covered by random product reviews on Amazon.com and random notes from an individual.

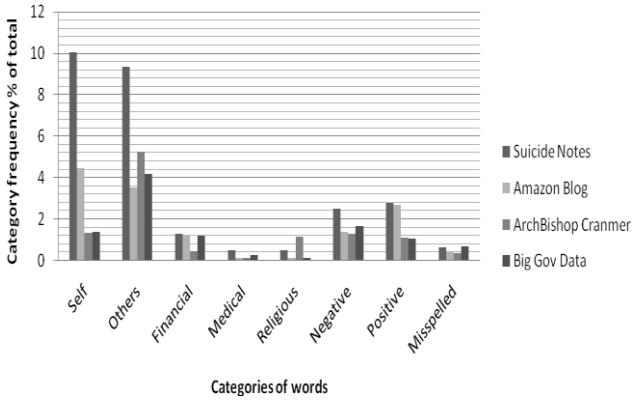


Figure 1. Word densities by percentage part 1

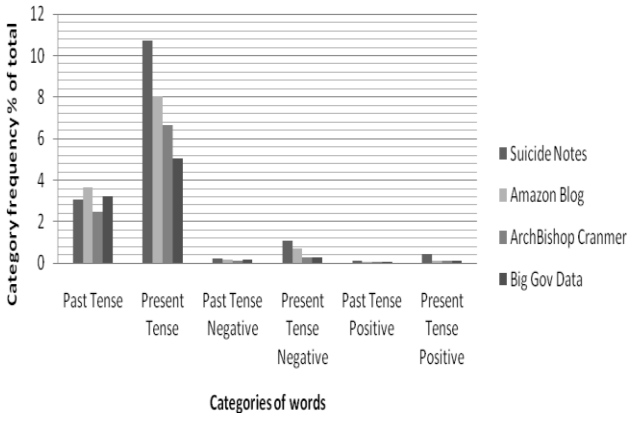


Figure 2. Word densities by percentage part 2

All of the words were grouped into abstract general categories and sorted in order from most frequently used to least frequently used words. Each grouping is defined by how dense they are by percentage compared to the entire dataset, i.e., how frequently each group is used in a given set of data. The categories used are references to self, others, financial terms, medical terms, religious terms, negative and positive words, and misspelled words. The densities of these categories are shown in Fig. 1. In addition to these categories, past tense and present tense words are also included along with their corresponding negative and positive references as shown in Fig. 2.

The results show that the greatest differentiation between the suicide notes and the non-suicide notes is found in the references to self and others in Fig. 1 and present tense in Fig. 2. These particular categories of frequency predominate in the set of suicide notes. This outcome means that there would be no nodes in the fuzzy cognitive map that would push the final result toward a non-suicidal classification if it was analyzing a non-suicidal case. Therefore, the patterns have to be extracted on a word by word basis.

3.2 Extracting patterns from specific words in written material

The three best places to gather words that can provide varying reliable patterns are the groups for self references, references to others, and present tense. These groups contain the most references compared to any other kind and, therefore, the words in these categories are most likely to be found in a random set of notes to be analyzed and classified as suicidal or non-suicidal. The densities for these words, however, are not based on how many of each word is used in the entire dataset but rather on how many of each word is used in the group it occupies.

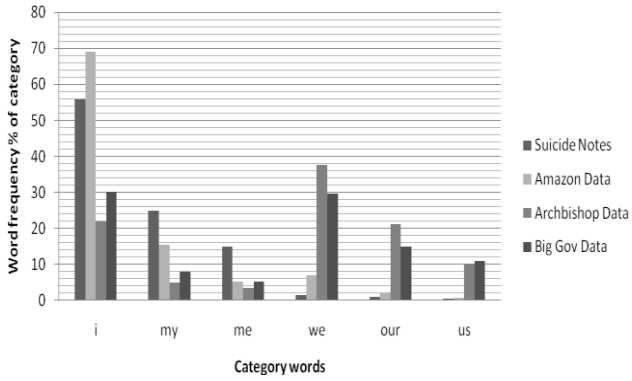


Figure 3. Word densities in self references by percentage

Upon further analysis of the three groups, there were a number of words that proved to have either distinct suicidal influences or distinct non-suicidal influences. All words that had small percentages over all four datasets or did not vary significantly between suicide and non-suicide were removed from consideration. Fig. 3 shows the final results for references to self.

On average each word has a specific affiliation to either the suicide notes or non-suicide notes. However, the Amazon.com data shows definite anomalies in the words "I," "we," "our," and "us" as compared with the other two non-suicide collection of notes. However, the apparent

pattern is that suicide notes have more singular self references, i.e. "I," "my," and "me," while non-suicide notes seem to have more group self references, i.e. "we," "our," and "us." Fig. 4 shows the results for references to others.

Again, the suicide notes contain a large amount of references to the word "you" whereas the non-suicide notes have larger references to "he," "they," "his," and "their." The Amazon.com data shows anomalies that are similar to the suicide data in the word "you" but has a great deal more influence in the word "they." The final results for present tense words are in Fig. 5.

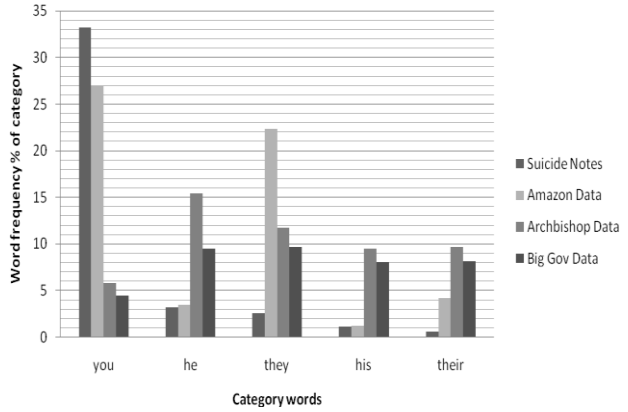


Figure 4. Word densities in others references by percentage

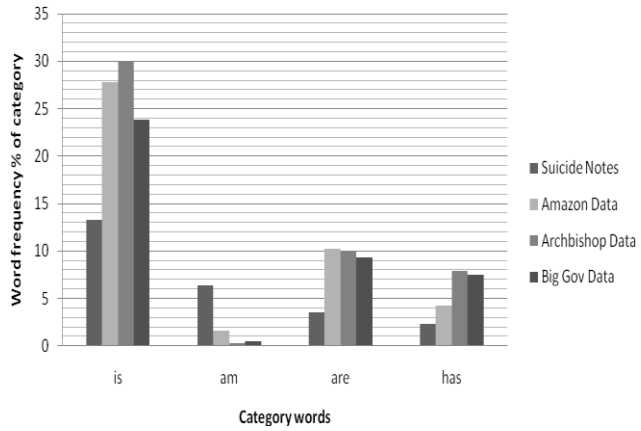


Figure 5. Word densities in present tense by percentage

In this group, the Amazon.com data acts similarly to the other non-suicide data except that the percentage for the word "has" is a little low, although not entirely problematic.

3.3 Developing the Initial FCM

The fuzzy cognitive maps consist of a series of connected nodes that will represent the words being used from Fig. 3-5. These words will in some way connect to a suicidal node that will determine the classification of the dataset. The simplest graph that can be constructed is for the suicidal node to be central with all word nodes connected to only that one node as shown in Fig. 6.

Each edge has a weight between -1.00 and 1.00 that determines how much influence and what kind of influence a particular node has on the suicidal node. The nodes on the left of Fig. 6 are all the nodes that are associated with suicide notes and thus have a positive influence, while all the nodes on the right represent non-suicide notes and are therefore negative in their influence.

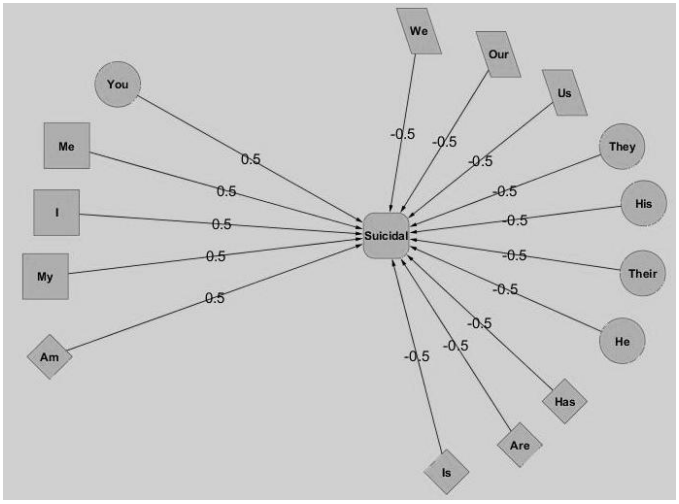


Figure 6. Initial fuzzy cognitive map

All of the initial weights start at 0.5 or -0.5. These settings make the assumption that all nodes have equal influence on the classification, although actual equality is unlikely. However, by starting at this assumption, it can be determined whether or not the general structure of the map accurately describes the causal relationships.

Each of the nodes starts at a particular value between 0.00 and 1.00 and then the graph is allowed to iterate in a computer program until the program either reaches a breaking point or until the graph reaches equilibrium. If the graph has reached equilibrium, then the final value of the suicide node is examined. If the value is over 0.80, i.e., over 80%, then the graph has determined the dataset to be suicidal. If the value is under 30%, then the dataset would be non-suicidal, and if the value is between 30% and 80%, then the classification is uncertain.

The starting values of the nodes are determined by normalizing the data in the particular group. For example, in Fig. 5, all four datasets would be normalized according to the archbishop result for the word "is." This means that about 30% is the new 100% to which all other values are compared to within that group. Fig. 3 and 4 would have their own number for normalization. The starting number for the suicidal node is 0.00 because it is assumed that there is no initial influence from this node.

The initial results for the fuzzy cognitive maps for each dataset were not entirely successful. The Amazon.com data in particular showed anomalies that made it similar to the

suicide notes. Therefore, in order to determine if this map structure can distinguish between the datasets correctly, additional techniques were employed to find an effective set of weights. By using machine learning techniques (supervised learning), a set of weights was discovered that allowed the fuzzy cognitive map to correctly classify each dataset. The graph with its final weights is shown in Fig. 7.

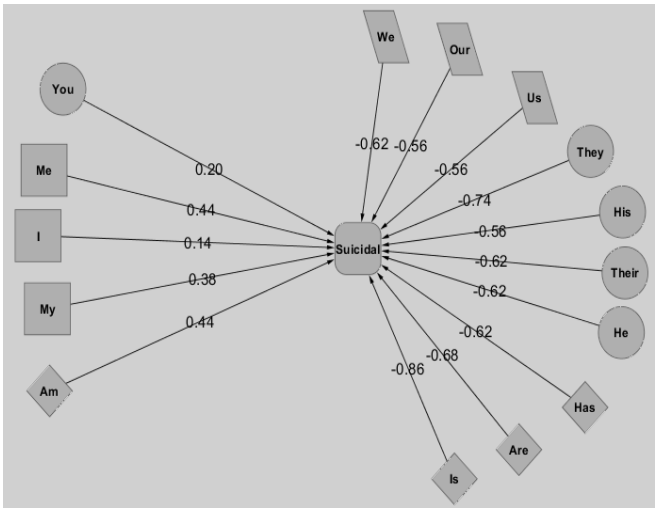


Figure 7. Final fuzzy cognitive map for testing

4 Testing the FCM

Now that a fuzzy cognitive map has been designed that can accurately classify the four datasets, this design must be tested against other collections of notes to see if the map can properly classify a random set of data.

Three more datasets were used for testing. These consisted of two sets of suicide notes and one non-suicide, each one only a fraction the size of the original four datasets. The first data set was a collection of suicide notes that contained a conglomeration of notes from the original suicide note collection as well as new notes. This dataset was obtained from the website www.well.com/~art/suicidenotes.html and is labeled as suicide notes 2 in the analysis. The second set was a collection of suicide notes from famous actors, poets, musicians, and celebrities in general. This set is labeled as suicide notes 3 in the analysis and was obtained from the website www.corsinet.com/braincandy/dying3.html. The final dataset was a collection of non-suicide notes from the private blog gregmankiw.blogspot.com.

4.1 Word Analysis of New Datasets

The final results for the word analysis are shown in Fig. 8, 9, and 10. As can be seen from the graphs, the three new datasets follow the same pattern for their respective classification with the exception of suicide notes 3. This dataset produced some anomalies in the form of large values in Fig. 10 for the words "is" and "are" that are similar to non-suicide patterns. Each of the new cases was normalized into the starting values for the nodes of Fig. 7. Each time, the fuzzy cognitive map accurately identified each dataset as either suicidal or non-suicidal. These findings suggest that this fuzzy cognitive map design is somewhat robust in that it was able to handle a random relatively small collection of suicide notes, i.e. suicide notes 3, and correctly identify them as such even considering the non-suicidal like components shown in Fig. 10.

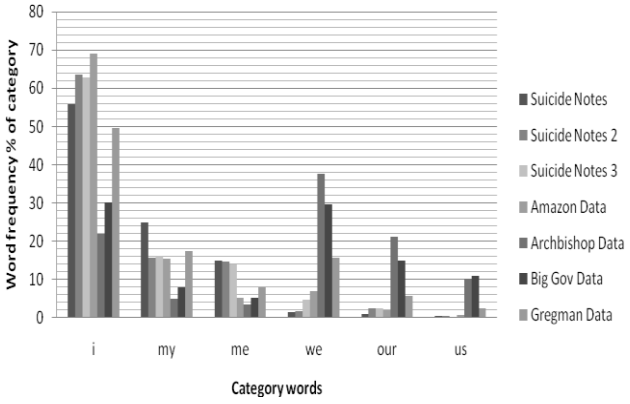


Figure 8. Word densities in self references by percentage

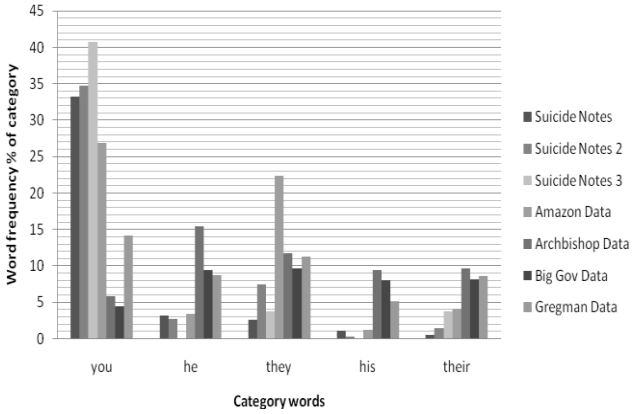


Figure 9. Word densities in others references by percentage

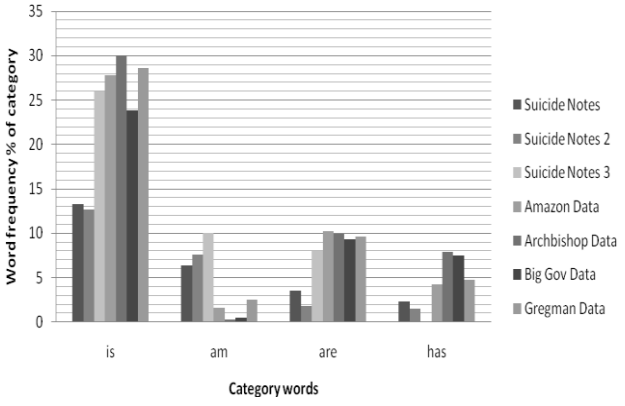


Figure 10. Word densities in present tense by percentage

4.2 Further Testing

One of the further tests done was an analysis based on gender distinction. A set of male and female notes were taken from the first set of suicide notes and separated to see if gender was a factor in word frequency patterns. In addition, a separate set of male and female suicide notes were taken from [3]. None of these four sets of notes showed any significant difference in word frequency patterns and were correctly identified by the fuzzy cognitive map in Fig. 7. Two non-suicide notes were then added. The first was extracted from a personal blog, colinhughes.blogspot.com, and was correctly identified as non-suicidal. The second collection was a set of personal letters from the 1820's that were written by medical patients to Dr. James Carmichael. These letters were part of a historical collection from the University of Virginia found at carmichael.lib.virginia.edu. The fuzzy cognitive map was unable to distinguish these notes from the suicide notes. By performing a principle component analysis (PCA) on the values of the nodes in the map, the suicide and non-suicide notes are shown clearly separate from one another with the exception of the Carmichael notes. These notes are found in the cluster of suicide notes on the bottom left of Fig. 11.

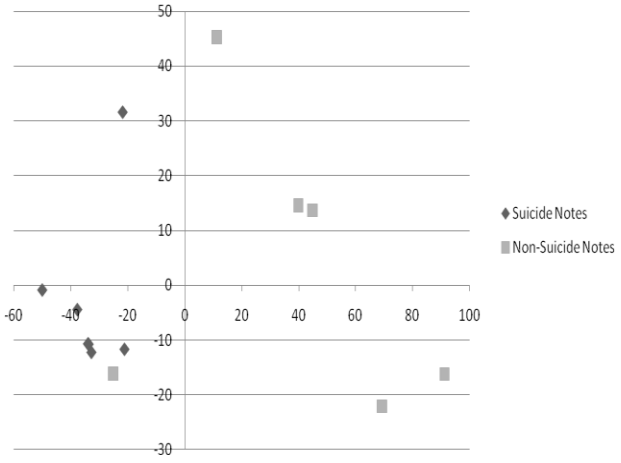


Figure 11. PCA of 13 notes

One possible explanation for the similarity of the Carmichael notes to the suicide notes is that they were written by patients self-focused on their physical suffering. This condition could cause particular word patterns to form similar to suicide notes also produced by suffering individuals. Another possibility is that suicide notes are often personal letters from one person to another. The Carmichael notes follow this same pattern of personal notes.

The problem was solved by adding more words from a new general category. Five more words were chosen from the past tense general category. The five words are "tried," "got," "told," "wanted," and "were." By adding these words, there were enough distinct word patterns that were able to separate the Carmichael notes from the cluster of suicide notes. All that was left to be done was to find a new set of weights that would be able to separate the suicide from the non-suicide notes. After redistributing the weights, the new fuzzy cognitive map, Fig. 12, was able to successfully identify the Carmichael notes as non-suicidal.

Finally, two more sets of personal letters were added to datasets for a total of 15. The first set was a collection of letters from Alexander Bell to his wife. These were most similar to the Carmichael data. The last dataset was a collection of personal letters from a soldier during the Spanish American War. Both sets came from the digital collection at the Library of Congress. The final results are shown in the PCA of the data, Fig. 13.

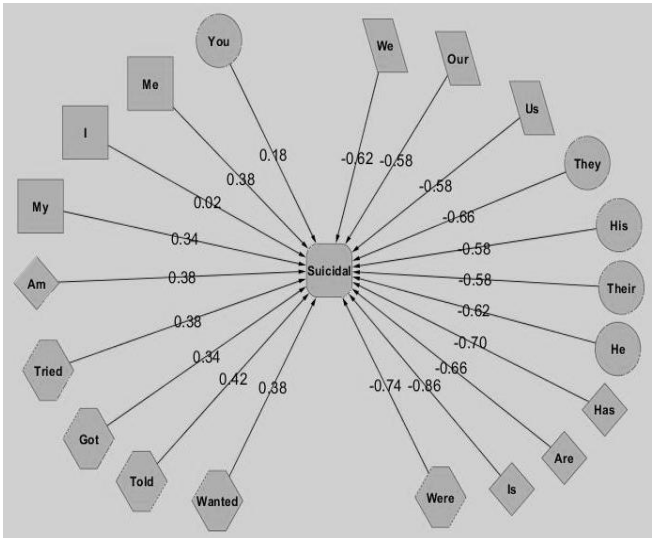


Figure 12. New fuzzy cognitive map

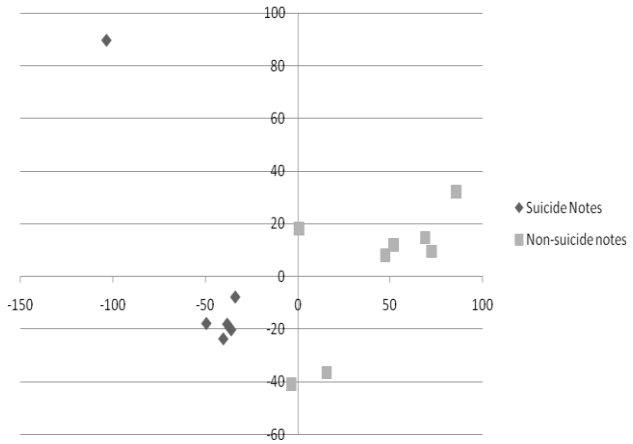


Figure 13. Final results for all 15 datasets

With the addition of the five past tense words, the Carmichael notes shifted away from the suicide cluster and are almost at the y-axis at y ≈ -40. The collection of Alexander Bell's letters appears to the right of the Carmichael notes. Finally, the Spanish American War

letters are found in the cluster on the right hand side of the graph near x ≈ 75.

5 Conclusion

The results of this research show the potential to differentiate between suicide and non-suicide writings by only analyzing word frequency patterns. By converting these patterns to input values for the fuzzy cognitive map, the datasets used in this research were successfully identified. However, further testing should be done in order to determine the usefulness of this approach in all given situations. Further tests need to be done on age based collections, since there is the possibility that word frequency patterns vary by age group. Another area of testing would be an analysis of suicidal ideation or intent to commit suicide [1] where a person talks of committing suicide but does not attempt suicide. Finally, further tests can be done from other collections of suicide and non-suicide notes that may result in further adjustments to the fuzzy cognitive map.

Some practical applications of this work could be found in a number of different areas. One possibility is with suicide prevention organizations that can use this method for tracking potential suicide victims who are seeking help, or for individuals who have already made a suicide attempt. In the latter case, the risk is higher that they might make another attempt and the proposed method of this work could be used to help identify suicidal trends that may result in another suicide attempt. Another area that can benefit is the military. According to Time magazine, there were more deaths in the military due to suicides than combat deaths in the entire Afghanistan War [20] up to the time the article was written. It might be possible to reduce the number of suicides by using the proposed method of this work to analyze communications from soldiers and identify any suicidal trends that might be occurring.

It is expected that the results of this research will further the understanding of causal relationships as they describe human behavior. Through this research, another technique for capturing causal relationships has been developed. Identification of causality could allow the ability to predict the consequences of actions from military strategies, governmental restructuring or societal rebuilding [2] [4]. In this instance, by using this fuzzy cognitive map technique, suicidal tendencies could be detected early to allow intervention that could save lives.

6 References

[1] S. Barnow and M. Linden, "Suicidality and Tiredness of Life Among Very Old Persons: Results from the Berlin Aging Study (BASE)," *Archives of Suicide Research*, 1997, pp. 171-182.

[2] B. Kosko, "Fuzzy Cognitive Maps," *Academic Press, Inc.*, vol. 24, 1986, pp. 65-75.

[3] A. A. Leenaars, *Suicide Notes Predictive Clues and Patterns*, Windsor Ontario, Canada: Human Sciences Press, Inc., 1988.

[4] L. Mazlack, "Approximate Representations in the Medical Domain," *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence*, Toronto, August 31 – September 3, 2010.

[5] A. S. Richardson, H. A. Bergen, G. Martin, L. Roeger and S. Allison, "Perceived Academic Performance as an Indicator of Risk of Attempted Suicide in Young Adolescents," *Archives of Suicide Research*, 9, 2005, 163-176.

[6] B. L. Paulson and R. D. Everall, "The Teen Suicide Research Project," *The Alberta Journal of Educational Research*, 47, 2001, 91-94.

[7] D. Leo, P. A. Hickey, G. Meneghel and C. H. Cantor, "Blindness, Fear of Sight Loss, and Suicide," *Psychosomatics*, 40, 1999, 339-344.

[8] M. Hagiwara, "Extended Fuzzy Cognitive Maps," *IEEE International Conference on Fuzzy Systems*, San Diego, CA, March 8-12, 1992, 795-801.

[9] B. Kosko, 1993, *Fuzzy Thinking: The New Science of Fuzzy Logic*, Hyperion, New York.

[10] G. D. Weber, "Discovering Causal Relations by Experimentation: Causal Trees," *Proceedings of MAICS*, 1997, 91-98.

[11] J. Pearl and T. S. Verma, "A Theory of Inferred Causation," *Second International Conference on the Principles of Knowledge Representation and Reasoning*, 1991, 441-452.

[12] S. Stromsten, "First-Order Causal Knowledge," 2003.

[13] D. Buchsbaum, A. Gopnik, T. L. Griffiths and P. Shafto, "Children's Imitation of Causal Action Sequences is Influenced by Statistical and Pedagogical Evidence," *Cognition*, In Press, Corrected Proof, 2011.

[14] K. Chan and W. Lam, "Extracting Causation Knowledge from Natural Language Texts," *International Journal of Intelligent Systems*, 20, 2005, 327-358.

[15] H. Sakaj, S. Sekine and S. Masuyama, "Extracting Causal Knowledge Using Clue Phrases and Syntactic Patterns," *Proceedings of the 7th International Conference on Practical Aspects of Knowledge Management*, Springer-verlag, Berlin, 2008, 111-122.

[16] R. Girju, "Automatic Detection of Causal Relations for Question Answering," *Proceedings of the ACL 2003 workshop on Multilingual Summarization and Question Answering*, 2003, 76-83.

[17] C. S. G. Khoo, S. Chan and Y. Niu, "Extracting Causal Knowledge from a Medical Database Using Graphical Patterns," *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, Hong Kong, Association for Computational Linguistics, Stroudsburg, PA, USA, 2000, 336-343.

[18] W. Buntine, "A Guide to the Literature on Learning Probabilistic Networks from Data," *IEEE Transactions on Knowledge and Data Engineering*, 8, 1996, 195-210.

[19] J. Pestian, H. Nasrallah, P. Matykiewicz, A Bennett and A. Leenaars, "Suicide Note Classification Using Natural Language Processing: A Content Analysis," *Biomedical Informatics Insights*, 2010, 19-28.

[20] M. Thompson, "Is the U.S. Army Losing Its War on Suicide," *Time*, 2010.

Autonomous Real-Time Site Selection for Venus and Titan Landing using Evolutionary Fuzzy Cognitive Maps

R. Furfaro[1], J. S. Kargel[2], and W. Fink[3]
[1]Department of Systems and Industrial Engineering, University of Arizona, Tucson, AZ, USA
[2]Department of Hydrology and Water Resources, University of Arizona, Tucson, AZ, USA
[3]Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ, USA

Abstract - *Future science-driven landing missions, conceived to collect in-situ data on regions of planetary bodies that have the highest potential to yield important scientific discoveries, will require a higher degree of autonomy. The latter includes the ability of the spacecraft to autonomously select the landing site using real-time data acquired during the descent phase. This paper presents the development of an Evolutionary Fuzzy Cognitive Map (E-FCM) model that implements an artificial intelligence system capable of selecting a landing site with the highest potential for scientific discoveries constrained by the requirement of soft landing on a region with safe terrain. The proposed E-FCM evolves its internal states and interconnections as function of the external data collected during the descent, therefore improving the decision process as more accurate information is available. The E-FCM is constructed using knowledge accumulated by experts and it is tested on scenarios that simulate the decision-making process during the descent toward the Hyndla Regio on Venus. The E-FCM is shown to quickly reach conclusions that are consistent with what a planetary expert would decide if the scientist were presented, in real-time, with the same available information. The proposed methodology is fast and efficient and may be suitable for on-board spacecraft implementation and real-time decision-making during the course of any robotic exploration of the Solar System.*

Keywords: Planetary Exploration, Planetary Landing, Autonomous Systems, Fuzzy Cognitive Maps

1 Introduction

Future unconstrained and science-driven NASA and ESA missions to explore planetary bodies in the Solar System will require soft landing in sites that have the potential to yield the highest geological and exobiological information. During the planning of any landing mission, scientists and engineers select an appropriate landing site on the planetary body of interest using data acquired during the reconnaissance of previously deployed robotic spacecrafts. The completeness of currently available information varies widely from body to body and depends critically on the number of missions deployed on the particular planet and/or natural satellite as well as on their geophysical properties (e.g. dense, thin or no atmosphere) which may inherently make the acquisition of surface data from orbiting spacecraft very difficult. For example, instruments on board spacecrafts currently orbiting Mars (e.g. Mars Reconnaissance Orbiter [1], Mars Odyssey [2]) are streaming a wealth of data about the red planet, thus providing a large amount of information to the scientists that can employ the available data to make the best possible landing site selection. On the contrary, other planetary bodies of high interest such as Venus and Titan have less available information and subsequently, a-priori landing site selection becomes more problematic. Due to a dense and opaque atmosphere which limits the electromagnetic bands available to passive optical instruments, both Venus and Titan have been mainly mapped using Synthetic Aperture Radar (SAR, e.g. Magellan SAR and Cassini SAR [3]) which generates images using the backscattered signal collected by the spacecraft antenna. SAR images have limited spatial resolution and they are harder to interpret, therefore making the ground-based landing selection process extremely difficult. Importantly, even in the case of Mars, a-priori selection of a suitable landing site is a complex process which involves the planetary science community at large. For example, NASA Jet Propulsion Laboratory has been organizing a series of workshops to actively engage Mars scientists to reach a consensus for the selection of the upcoming Mars Science Laboratory (MSL) landing site [4].

Whereas ground-based pre-planning and landing site selection is an important activity for any space mission comprising a lander, the final and best landing site selection may be possibly executed in real-time during the Entry, Descent and Landing (EDL) phase. Here, we define "best" as the landing site that has the potential to yield the highest potential for scientific discoveries and that satisfies specific landing safety constraints. Real-time, autonomous selection of such a site requires that the robotic lander is equipped with a system capable of a higher degree of autonomy. Such a system should (1) include software packages that enable fully automated and comprehensive identification, characterization, and quantification of features information within an operational region with subsequent target prioritization and selection for close-up reexamination (e.g. Automated Global Feature Analyzer, AGFA [5]); and (2) integrate existing information with acquired, "in transit" spatial and temporal sensor data to automatically perform intelligent planetary landing, which includes identification of sites with the highest

potential to yield significant geological and astrobiological information ([6],[7]) .

In this work, we design and simulate an advanced intelligent system capable of autonomously selecting landing sites using data coming in real-time from the lander sensors. We use Evolutionary Fuzzy Cognitive Maps (E-FCM, [8]) to model the cognitive reasoning of a planetary scientist that is presented with the same information available to the lander on-board computer and decides in real-time where to drive the spacecraft for safe landing. While a large numbers of techniques in the AI domain are available (e.g. fuzzy experts [9],[7]) , the E-FCM for real-time landing decision making was selected because of its ability to model complex processes comprising a large number of interacting parameters. Importantly, available experience and knowledge accumulated by planetary scientists can be easily translated in a cognitive map that is expressed through the use of concepts connected via causal relationships. Moreover, the ability of the proposed algorithm to reach conclusions in a fast and efficient way, make it ideal for real-time implementation on the spacecraft on-board microprocessor.

To our knowledge, FCMs have been only recently employed to design algorithms for autonomous interpretation of planetary data (see [10]). The main goal of this work is to show how FCMs in general, and E-FCMs in particular, can be used to advance the state of the art of autonomy in planetary exploration by providing an effective inference platform that can be easily understood by planetary scientists.

2 Autonomous Landing on Planetary Bodies

2.1 Landing on Venus and Titan: the Need for Autonomous Systems

Landing on planetary bodies with dense atmosphere such as Titan and Venus is extremely challenging. In any future science-driven and unconstrained landing scenarios, including soft precision landing (< 100 m) and pin-point landing (<1 m), autonomy will play more and more a critical role to a) provide autonomous selection of the landing site based on real-time data, b) implement a targeting program that will generate a flyable trajectory to the selected target and c) execute real-time guidance algorithms to drive the system to the desired location. Current flight-ready technology has been effective to land a spacecraft on a preselected region within a landing ellipse of 120x20 km (e.g. Phoenix Mission to Mars [11]). Future missions (e.g. MSL, [12]) have the potential to shrink the landing ellipse to less than 10 km. Clearly, autonomous, real-time landing site selection has never been implemented. The latter is extremely desirable because the selection takes place in real-time by reasoning on data collected during the descent to determine sites with the highest potential of scientific discoveries and avoid areas with high hazard potential.

In human history, only one landing probe has been delivered to the Titan surface, i.e. the Huygens probe, which landed on the Titan surface on January 14, 2005 [13]. The descent was completely preplanned and unguided due to scarce surface information. The probe landed on what is now interpreted to be a methane-rich outflow channel. During the descent, the camera system collected a set of images showing a plethora of interesting geological features which may have constituted interesting landing sites. Landing on Venus has been attempted many times as documented by the results from Venera and Vega missions [14]. Venus surface has been mapped using the Magellan SAR which is the only mean to probe the surface beyond its dense atmosphere. Figure 1 shows the landing ellipse of the region selected by Venera 10, which is also the region selected to test our design (see section 4). The region presents 8 stratigraphic units ([15]) and, for our analysis, has been subdivided in three regions. Region #1 is mainly comprised of what is classified as "Tesserae Units", i.e. ancient terrains comprising dislocated units of tectonic origin. Region #2 is mainly comprised of highly fractured plains that exhibit lobate flow field features. Region #3 exhibits a smooth (radar-dark) behavior interpreted as volcanic planes comprising materials that resembles to terrestrial volcanic rocks ([16]). In all Venera and Vega missions, the landing site has been preselected as well.

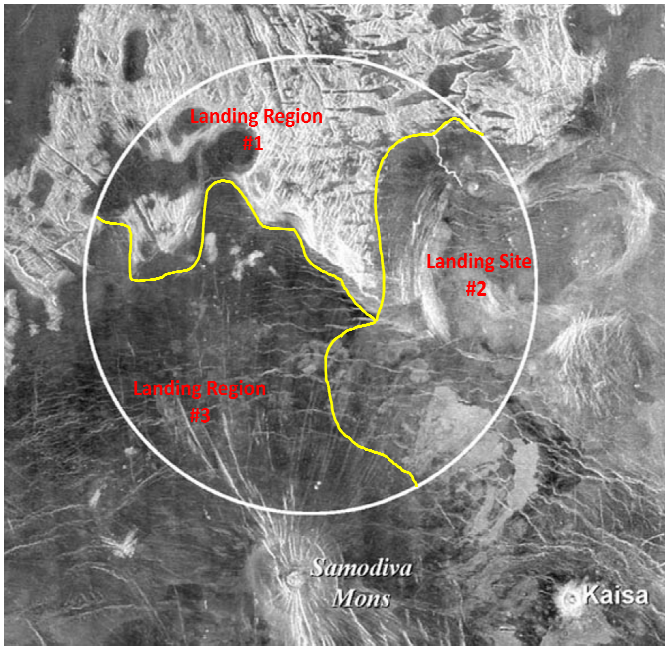


Figure 1: Venera 10 Landing ellipse. The area has been divided in three regions with features (real and hypothesized) that are used to test the E-FCM algorithm

2.2 Landing on Venus and Titan: The need for Autonomous Systems

Evolutionary Fuzzy Cognitive Maps (E-FCM, [8],[17]) may be considered as an extension of FCMs specifically developed to model variable internal states as well as the dynamic,

complex and causally related context variables (i.e. coming from data acquired during the landing descent). In its basic formulation, FCMs are digraphs designed to capture the cause/effect relationships exhibited by a system ([18]). Two basic elements form the backbone of the maps, i.e. nodes and arcs. Nodes are the concepts representing factors and attributes of the modeled system, e.g. inputs, outputs, states, variables, events, goals, as well as trends. Arcs are introduced to describe the causal relationships between concepts with a degree of causality. Extending this approach further, in E-FCMs, the concept states evolve in real-time as function of the internal mental state, external inputs and possibly external causalities.

Table 1: Concepts Description and Concepts Values

Concepts	Concept Description	Value
C1, C12, C23: Landing Region #1,#2,#3	Index for Landing Site Selection	Continuous Value [0,1]
C2, C13, C24: Potential for Scientific Discoveries (PSD 1,2,3)	This concept indicates the potential exhibited by the region to yield significant discoveries	Five fuzzy values, {VL,L,M,H,VH}
C3, C14, C25: Potential for Hazards (PHZ 1,2,3)	This concept indicates the regional potential for landing hazards	Five fuzzy values, {VL,L,M,H,VH}
C4, C15, C26: Volcanic Plain Features (VPF 1,2,3)	VPF indicates the level of presence for features associated to volcanic landforms	Six fuzzy values, {A,VL,L,M,H,VH}
C5, C16, C27: Flow-Like Features (FLF 1,2,3)	FLF indicates the level of presence for features volcanic flows	Six fuzzy values, {A,VL,L,M,H,VH}
C6, C17, C28: Ancient Terrain Features (ATF 1,2,3)	ATF indicates the level of presence for features associated with ancient terrains	Six fuzzy values, {A,VL,L,M,H,VH}
C7, C18, C29: Smooth Terrain (ST 1,2,3)	ST indicates the level of smoothness of the region	Five fuzzy values, {VL,L,M,H,VH}
C8, C19, C30: Rough Terrain (RT 1,2,3)	RT indicates the level of roughness of the region	Five fuzzy values, {VL,L,M,H,VH}
C9, C20, C31: High Slope Terrain (HST 1,2,3)	HST indicates the high slope level of the region	Five fuzzy values, {VL,L,M,H,VH}
C10, C21, C32: Low Slope Terrain (LST 1,2,3)	LST indicates the level of low slope level of the region	Five fuzzy values, {VL,L,M,H,VH}
C11, C22, C33: Featureless Terrain (FT 1,2,3)	FT indicates the level of featureless terrain of the region	Six fuzzy values, {A,VL,L,M,H,VH}

Indeed, each concept is equipped with its own update schedule as well as subjected to a small self-mutation probability. Moreover, the causal connection between concepts is fired according to a specified conditional probability. The evolutionary extension to FCMs has been chosen due to the nature of the landing selection process. During the Entry, Descent and Landing (EDL) phase, data are continuously streamed in real-time to the spacecraft computer and they are processed to determine the landing site using all available information. However, such data are remotely collected and subjected to uncertainty due to the limited resolution of the instrumentation. With the assumption that as the lander gets closer to the targeted region, more accurate information is available, the E-FCM must have the ability to adapt and dynamically update the connections strength to account for newly available data. In this case, differently than conventional FCMs, concepts are represented by tuples of properties, i.e. $\mathbf{A} = [\mathbf{A}_V, \mathbf{T}, \mathbf{PS}]$. Here, $\mathbf{A}_V$ denotes the fuzzy values of the concepts **C** (same as FCM). Concepts values generally range between [-1, 1] or [0,1]. For a system of N concepts, $\mathbf{A}_V = [A_{V1}, A_{V2},\ldots,A_{VN}]$ is an evolving vector that represents the time evolution of the each individual concept value C_i. $\mathbf{T} = [T_1,T_2,\ldots T_N]$ is a Nx1 vector that represents the evolving time schedule of each concept. The latter accounts for the fact that various concepts may have a different real-time update schedule. $\mathbf{P}_S = [P_{S1},P_{S2},\ldots,P_{SN}]$ is the state mutation vector which account for the possibility that each concept may randomly alter its internal state in real time. According to Cai et. al. ([8]), the self-mutation probability must be modeled as small value to avoid system's instability.

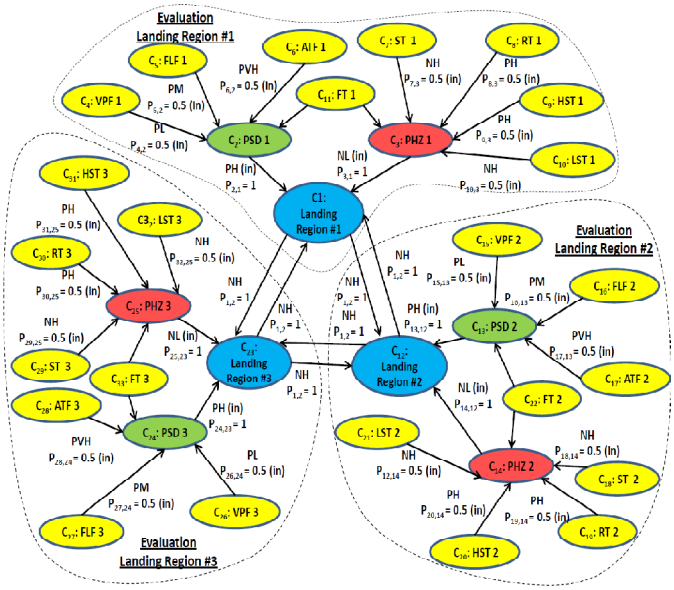


Figure 2: E-FCM topological structure. The weights are defined using fuzzy linguistic values (VL = 0.1, L = 0.25, M = 0.5, H = 0.75, VH = 0.9). P and N indicate respectively direct and inverse connection. Initial values of probabilities are indicated as well.

The causal relationship between concepts is defined as the tuple $\mathbf{R} = [\mathbf{W}, \mathbf{Pm}]$ which extend the conventional FCM approach (which uses only the fuzzy connection matrix) to allow the incorporation of uncertainty via fuzziness and randomness. For a system of N concepts, $\mathbf{W} = \{w_{ij}\}$ is the NxN weight matrix that represents the causal relationship between two concepts in fuzzy terms (e.g. high, low, medium, absent, etc.). The connection can be either positive or negative. $\mathbf{P}_m = \{p_{i,j}\}$ is the NxN matrix of causal probabilities between the interconnections. Causal probabilities may also be described in fuzzy terms. They represent the uncertainty of connections between concepts. For example, $p_{i,j}$ represent the probability that the concept A_i influences A_j with the strength

defined by the correspondent w_{ij}. The value A_i of the concept C_i is computed by accounting for all possible influences deriving from interconnected concepts as well as the causal probabilities deriving from $\mathbf{P}_m$. At each time interval and according to the predefined time schedule **T**, the concepts are updated using the following formula:

$$\boldsymbol{A}_i(k+1) = f\left(\boldsymbol{A}_i(k) + \sum_{\substack{i=1,N \\ i \neq j}} \boldsymbol{A}_j(k) W_{ji}\right) \qquad (1)$$

Here, f(·) is the activation function used to regulate the state variable (bivalent, trivalent or logistic). The weight matrix is evaluated according to the causal probability matrix and the value is randomly changed according to the self-mutating probability.

The development and construction of the E-FCM is executed using knowledge accumulated by field experts who define the type and numbers of concepts, the strength of the connections and also the causal probability matrix. Various methodologies are available to define the connections strength. Papageorgiu et. al. ([19]) proposed to define the strength of the connection between concepts using fuzzy IF-THEN rules in the following fashion:

IF value of concept C_i is B THEN value of concept C_j is C and the linguistic weight w_{ij} is E

Here, B,C,E are linguistic fuzzy values determined via appropriate membership functions with values in the range [0,1] for direct connection and [-1,0] for inverse connection. Indeed, for any of the established concepts, the experts determine the negative or positive effect of one concept on the others with a fuzzy degree of causation. As shown by [19], experts' opinions can be accounted individually via independent linguistic rules that are subsequently aggregated and de-fuzzified. Our group agreed on the structure of the map so that only one linguistic rule is defined to infer the fuzzy causal connection between concepts (i.e. fuzzy aggregation is not required).

3 E-FCM Methodology for Landing Decisions Making: the case of Venus

The problem of autonomous selection of a landing region that integrates published knowledge and real-time acquired information on the journey toward the planetary surface is a complex process that requires interaction between large numbers of parameters. Moreover, for a given scenario, defining the criteria that allow a clear definition of what is the region that yields the maximum possible scientific information is a matter of debate within the planetary science community. Here, we consider the case of landing on Venus and we focus on designing an E-FCM that selects the landing region, among the observables, that exhibits the terrain with the most ancient features and hence has the potential of unfolding a large portion of the geologic history of the planet ([17]). However, potential for scientific discoveries and geological understanding of any of the considered regions must be consistent with the ability of the spacecraft to safely land on the selected site. Because of the limited SAR resolution and the difficulty for its correct image interpretation, a-priori analysis of potential Venusian landing regions may not unfold critical features that may yield higher (or lower) potential for scientific discoveries and/or potential hazards. Thus, the overall goal is to construct an E-FCM that ingests data during portions of the EDL and, for the pre-selected regions, it infers what is the potential for scientific discoveries (in the sense clarified above) and potential for hazards. Such indicators are then used by the map to autonomously select the best site for a safe soft landing. The proposed E-FCM evolves in time according to a prescribed schedule to a) account for data streamed into the system from the sensors and b) adapt for reasoning under uncertainty and ambiguity of the data available as function of time. Our team designed an E-FCM model that accounts for 33 interconnected concepts (see table 1). The selected concepts, which have been linked to data that may be determined using real-time feature extraction software (e.g. AGFA, [5]) are divided in three major groups. Each group contains input concepts that have been selected to account for features that are required to infer both the potential for scientific discoveries and the potential for hazards for any of the three regions located within the selected landing area on the southern part of the Venusian Hyndla Regio (Figure 1). Indeed, the E-FCM is asked to select one landing site among the three pre-identified landing regions using hypothesized data acquired during a portion of the descent flight. For each of the three regions, a group of 4 variables are identified to influence the potential for scientific discoveries (e.g. C_4, C_5, C_6, C_{11} for landing region #1) whereas a group of 5 variables are shown to influence the potential for hazards (e.g. C_7, C_8, C_9, C_{10}, C_{11} for landing region #1). Both potential for scientific discoveries and potential for hazards influence the landing site selection. The weight matrix is selected by our team's planetary experts. For example, the link between concept C_6 (Ancient Terrain Features for Landing Site #1) and C_2 (Potential for Scientific Discoveries on Landing Site #1) is established to be "Positive Very High (NVH)" or using an IF-THEN formalism:

IF a small change in the value of concept C_6 occurs THEN a very high change in the value of the concept C_2 is caused. Inference: the influence of C_6 on C_2 is Positive Very High.

Conversely, the influence between the value of the concept C_7 (Smooth Terrain in landing Site #1) and C_3 (Potential for Hazards on Landing Site #1) is "Negative High (NH)" or:

IF a small change in the value of concept C_7 occurs THEN a negative high change in the value of the concept C_3 is caused. Inference: the influence of C_7 on C_3 is Negative Very High.

In absolute terms, the influence of the potential for scientific discoveries and hazards on the landing region concepts has a

value of PH and NH, respectively. However, the connection is selected to be time dependent in the following sense: During the descent toward the targeted area, the influence of the potential for scientific discovery is initially set to be PH and the potential for hazard is initially set to be Negative Low (NL). As the lander gets closer to the surface, the influence of the hazard on the landing site selection is increased (limit to NH when the final decision is made) whereas the influence of the potential for scientific discoveries is reduced (limit to PL when the final decision is taken).

Table 2: Ground Truth for the Landing Scenario #1

Scenario # 1	**Landing Region #1**	**Landing Region #2**	**Landing Region #3**
VPF	L	L	VH
FLF	L	VH	L
ATF	H	A	A
ST	H	H	H
RT	L	L	L
HST	L	L	L
LST	H	H	H
FT	VL	VL	VL

Table 3: Ground Truth for the Landing Scenario #2

Scenario #2	**Landing Region #1**	**Landing Region #2**	**Landing Region #3**
VPF	L	L	VH
FLF	L	VH	L
ATF	H	A	A
ST	L	H	H
RT	H	L	L
HST	H	L	L
LST	L	H	H
FT	VL	VL	VL

The goal is to ensure that priority is given to landing safety especially when less uncertain, higher resolution data are available. The probability matrix is also established by our field experts and, for certain connections, it is assumed to be time-dependent. As a general guide, the probability of connection between input data (e.g. concepts C_4-C_{11}) and derived concepts (e.g. C_2, C_3) is assumed to increase with the time of flight to account for the fact that uncertainty in data is less pronounced and therefore the connection is more probable. Figure 2 illustrates the designed E-FCM model for landing site selection with numerical values for weights and causal probability. Importantly, the map topology has been selected such that the there is an internal competition to select the winning landing site.

4 Simulations and Results

After the construction of the E-FCM, a number of scenarios have been considered to simulate the behavior of the algorithm. It is assumed that the scientific team responsible for soft landing on Venus, selects the southern part of Hyndla Regio as landing area (see figure 1). The area is subdivided in three landing sites for which the "ground truth" is assumed to be known and established a-priori. The EDL trajectory has been designed such that the lander can autonomously select the landing site using the E-FCM before t_F, time after which guidance constraints impose that the other two regions are outside the reachability domain. It is also assumed that the spacecraft is able to collect data for 120 seconds before t_F is achieved.

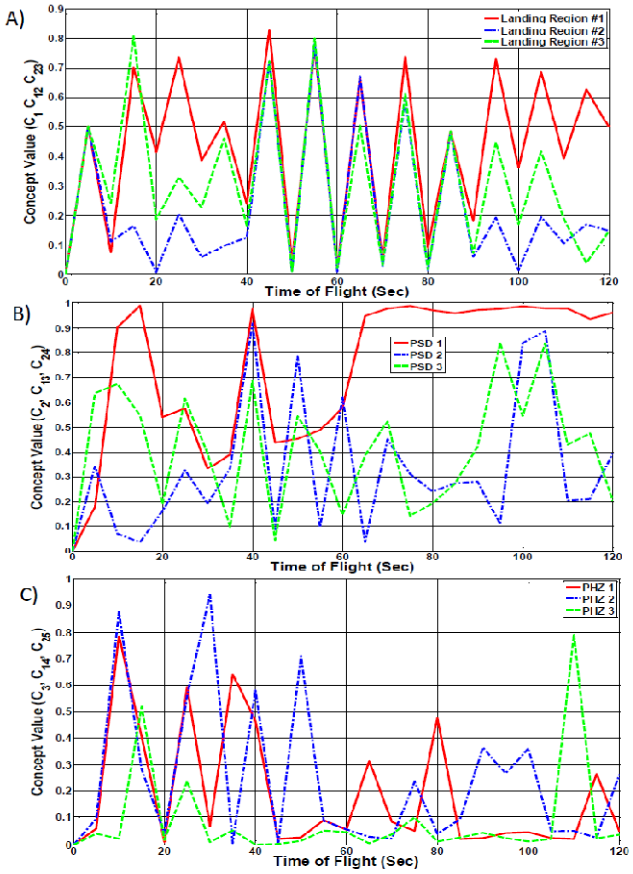


Figure 3: Landing scenario #1 E-FCM time-dependent output for selected concepts. A) Landing selections values; B) PSD values; C) PHZ values.

The information is updated each 5 seconds during which the lander acquires and processes images to extract the features that are input to the E-FCM. For any of the established scenarios, it is assumed that the spacecraft collects data with an uncertainty that is function of the flight time. More specifically, the "ground truth" is set to be the mean value of a data sampling Gaussian distribution with a standard deviation that decrease as t_F is approached (i.e. as the lander

gets closer to the surface, the improved instrument resolution yields more accurate data). With this setting, data are continuously updated and the E-FCM run to infer, at each given time interval, what is the best landing region. Finally, it is noted that all concepts are updated synchronously (same time schedule) and no self-mutation probability is implemented. The following two scenarios are considered.

Scenario #1: For the first scenario, the "ground truth" is constructed using fuzzy linguistic values as reported in table 2. In this case, the available hypothesized data show a region #1 that is the most attractive for landing. Indeed, the large presence of ancient terrain features makes the region more attractive from the prospective of unfolding the ancient geological history of Venus. All regions are shown to be very safe safe for landing (plenty of smooth and low slopes terrains). Landing regions #2 and #3 have also flow-like features and volcanic terrains which may be of scientific interest but lower priority. It is therefore expected that the E-FCM selects Landing Region #1 as landing scenario. The simulation is initiated by setting up an initial concept value vector whose input values are assigned using a normal (Gaussian) distribution. The E-FCM evolves both weights and concepts values with input values updated each five seconds. The simulation results are reported in figure 3 which shows the time evolution of landing values as well as PSD and PHZ for all regions. As evident from figure 3A, the highest value is reached by region #1 which wins the competition with the other two regions. As the acquired data increasingly indicate that all regions are safe for landing, the E-FCM chooses the region with the highest potential for scientific discoveries, consistently with our expert analysis based on ground truth data.

Scenario #2: The second scenario is similar to the first one. As reported in the ground truth table (see table 3), landing region #2 and #3 are identical to the first scenario. Landing region #1 still shows a high presence of ancient terrains which makes it attractive from a scientific discovery point of view. However, the region now indicates a very robust presence of rough terrain and high slope surface which is a strong indication of potential for hazards which should discourage the landing selection. Figure 4 shows that the E-FCM reached the conclusion that region #1 has both highest PSD and PHZ and subsequently disregard it for landing selecting landing region #2 which ahs lowest scientific interest but exhibits safer landing terrains.

5 Conclusions

The artificial intelligence approach used in this work focuses on developing evolutionary fuzzy cognitive techniques that mimics the planetary scientist selection process for landing site selection, with special emphasis on Venus and Titan. It is shown that the proposed E-FCM reaches the same conclusions as field experts and it is fast enough to be suitable for real-time, on-board implementation. The outlined methodology has the potential to be the premiere AI choice for cognitive reasoning on data for planetary exploration.

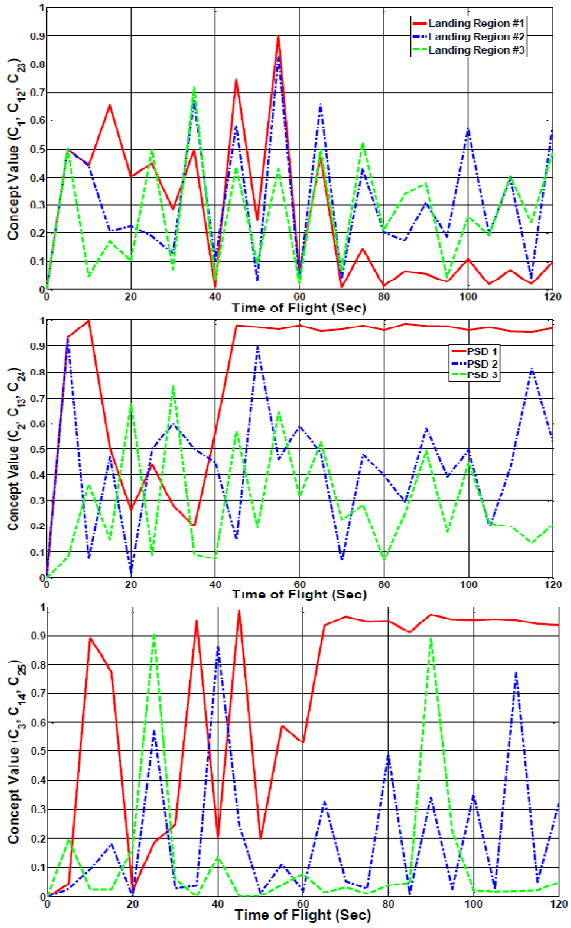


Figure 4: Landing scenario #2 E-FCM time-dependent output for selected concepts. Top: Landing selections values; Middle: PSD values; Bottom: PHZ values.

6 References

[1] Graf, J., et al. (2005), "The Mars Reconnaissance Orbiter Mission." *Acta Astronautica*, vol 57, pp. 566-578, 2005.

[2] Saunders, R., et al. (2004), 2001 Mars Odyssey mission summary, Space Sci. Rev., 110, 1 – 36.

[3] Elachi, C., et. al., (2005), Cassini Radar views the surface of Titan, Science, 308, 970–974.

[3] http://marsoweb.nas.nasa.gov/landingsites/msl/ /4th_workshop/program.html

[5] Fink W, Datta A, Dohm JM, Tarbell MA, Jobling FM, Furfaro R, Kargel JS, Schulze-Makuch D, Baker VR , (2008) Automated Global Feature Analyzer (AGFA) – A Driver for Tier-Scalable Reconnaissance; *IEEE Aerospace Conference Proceedings*, paper #1273; DOI: 10.1109/AERO.2008.4526422

[6] Fink, W., Dohm, J., M., Tarbell, M., A., Hare, T., M., Baker, V., R., (2005). Next-Generation Robotic Planetary Reconnaissance Missions: A Paradigm Shift. *Planetary and Space Science*, 53, 1419-1426.

[7] Furfaro R, Dohm JM, Fink W, Kargel JS, Schulze-Makuch D, Fairén AG, Ferré PT, Palmero-Rodriguez A, Baker VR, Hare TM, Tarbell M, Miyamoto HH, Komatsu G, (2008)The Search for Life Beyond Earth Through Fuzzy Expert Systems; *Planetary and Space Science*, Volume 56, Issues 3-4, 448-472.

[8] Yundong Cai, Chunyan Miao, Ah-Hwee Tan, Zhiqi Shen, Boyang Li, "Creating an Immersive Game World with Evolutionary Fuzzy Cognitive Maps," IEEE Computer Graphics and Applications, vol. 30, no. 2, pp. 58-70, Mar./Apr. 2010, doi:10.1109/MCG.2009.80.

[9] Furfaro R, Dohm J., M., Fink W., (2006), Fuzzy Logic Expert System for Tier-scalable Planetary Reconnaissance; *9th International Conference on Space Operations, AIAA*, Rome, Italy, June 19-23, 2006.

[10] Furfaro, R**.**, Kargel, J., S., Lunine, J., I., Fink, W., Bishop, M. P., (2010), Identification of Cryovolcanism on Titan Using Fuzzy Cognitive Maps, Planetary and Space Science, Volume 5, Issue 5, Pages 761–779.

[11] Shotwell, R, 2005, Phoenix—the first Mars Scout mission, *Acta Astronautica*, Volume 57, Issue 2-8, p. 121-134.

[12] Steltzner, A. D., Kipp, D. M., Chen, A., Burkhart, P. D.,Guernsey, C. S., Mendeck, G. F., Mitcheltree, R. A., Powell, R. W., Rivellini, T. P., San Martin, A. M., Way, D. W., 2006, Mars Science Laboratory Entry, Descent, and Landing System, *IEEE Aerospace Conference* Paper No. 2006-1497, Big Sky, MT, Mar. 2006.

[13] Lebreton, J.-P. *et al.* (2005), An overview of the descent and landing of the Huygens probe on Titan. *Nature* 438**,** 758–764 .

[14] Basilevsky, A. T., Ivanov, M. A., Head, J. W., Aittola, M., & Raitala, J. (2007), Landing on Venus: Past and future, Planet. Space Sci., 55, 2097.

[15] Abdrakhimov, A.M., 2005. Geology and geochemistry of the Venera 8, 9, 10, 13, 14, Vega 1, 2 landing sites. PhD Dissertation, Vernadsky Institute of Geochemistry and Analytical Chemistry, RAS, Moscow, 143pp.

[16] Kargel, J.S., Komatsu, G., Baker, V.R., Strom, R.G., 1993. The volcanology of Venera and VEGA landing sites and the geochemistry of Venus. Icarus 103, 253–275.

[17] Y. Cai, C. Miao, A.-H. Tan, and Z. Shen, "Context modeling with evolutionary fuzzy cognitive map in interactive storytelling," in *IEEE International Conference on Fuzzy Systems, WCCI 2008*, Hongkong, China, 2008, pp. 2320–2325.

[18] B. Kosko, "Fuzzy cognitive maps," *International Journal of Man Machine Studies*, vol. 24, pp. 66–75, 1986.

[19] Papageorgiou, E. I., Spyridonos, P., Ravazoula, P., Stylios, C. D., Groumpos, P. P., & Nikiforidis, G. (2006). Advanced soft computing diagnosis method for tumor grading. In Artificial intelligence in medicine (Vol. 36(1), pp. 59–70).

Importance of Factors Effective upon Consumers' Perception of Fairness in Dynamic Pricing: An FCM Approach

SUBMITTED TO: ICAI

Authors:
Yashar Dehdashti
MBA student, Graduate School of Management and Economics, Sharif University of Technology, Tehran, Iran
yashar.dehdashti@gmail.com
Nooshin Lotfi
MBA student, Graduate School of Management and Economics, Sharif University of Technology, Tehran, Iran
nooshin.lotfi@gmail.com
Aidin Tajzadeh Namin
Marketing PhD student, School of Management, University of Texas at Dallas, Dallas, USA
aidin.tajzadehnamin@student.utdallas.edu
Dr. Manoochehr Najmi
Associate professor, Graduate School of Management and Economics, Sharif University of Technology, Tehran, Iran
m.najmi@sharif.ir

Keywords:
Transaction characteristics, price fairness judgment, consumer purchase satisfaction, fuzzy cognitive maps

Abstract
This article build upon *fairness in pricing* – i.e. a perceived fairness judgment by a buyer of a seller's prices (Haws and Bearden, 2006) - and *dynamic pricing* - a strategy in which prices vary over time, consumers, and/or circumstances (Haws and Bearden, 2006) - an analysis of the importance of the factors influencing consumers' perception of fairness. Specifically, it uses fuzzy cognitive maps on the model proposed by Haws and Bearden (2006) to find the most important path leading to consumer satisfaction with the price. The FCM analyzes the responses of a group of 30 experts from 4 leading companies in medical equipment importing to find the most important paths.

Introduction

Price fairness and dynamic pricing have been investigated from different perspectives. Moreover, many articles have addressed the factors important in dynamic pricing (Haws and Bearden, 2006). However, the importance of the factors involved has not been surveyed enough. Specifically, there are no definitive guidelines for managers as to which of the factors analyzed in the literature are most conducive to the final decisions of the managers. This article utilizes fuzzy cognitive maps to weigh the importance of factors mentioned by Haws and Bearden, 2006. The FCM analyzes the responses of a group of 30 experts from 4 leading

companies in medical equipment importing to find the most important paths. Upon finding the most important path leading to satisfaction with price, it proposes implementation methods and further research for future studies.

Price fairness refers to a perceived fairness judgment by a buyer of a seller's prices. Although fairness is a difficult concept to define, the perception of price fairness is part of a broader judgment of the overall merits of a deal (Haws and Bearden, 2006). Despite the importance of perceived price fairness, research on the topic has not been immense. Relevant research that does exist has largely been inspired by the principle of dual entitlement (Kahneman, Knetch, and Thaler 1986b), which argues that fairness perceptions are governed by the belief that firms are entitled to a reference profit and customers are entitled to a reference price (Bolton, et al. 2003).

Dynamic pricing is defined as a strategy in which prices vary over time, consumers, and/or circumstances. Elmaghraby and Keskinocak (2003) distinguish between two dynamic pricing models: price-posted mechanisms and price-discovery mechanisms. With price-posted mechanisms, frequent price changes are offered as "take-it-or-leave-it" prices, that is, the company is still in charge of setting the price. With price-discovery mechanisms, such as eBay, Priceline, or similar negotiated approaches, consumers have input into setting the final price.

Building upon previous researches in the area of dynamic pricing and consumer perceptions of fairness, Haws and Bearden (2006) have proposed the following model.

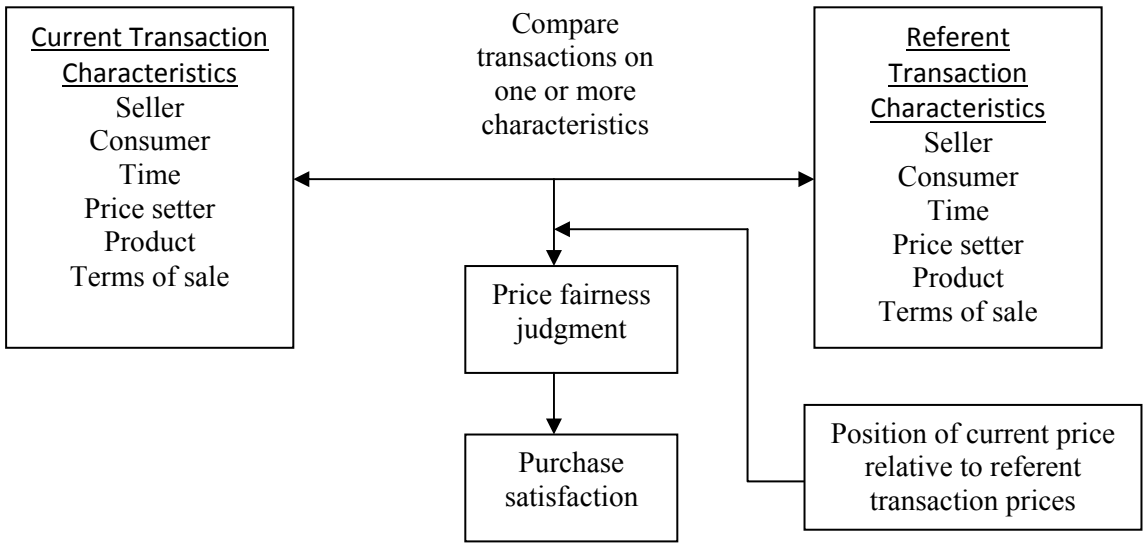


Figure 1-Transactoion characteristics and consumer judgments of price fairness

Seller

The seller affects the consumer's perception of price fairness when the seller is already familiar to the consumers or when the seller possesses a brand image conveying a sense

of pricing based on either better economies or stronger occupational ethics.

Terms of sale

Factors such as conditional payments, number of installments, etc. have been shown to be effective upon the consumer's perception of price fairness. The lower the installment rates and the more flexible the payment terms are, the higher the perception of price fairness will be on the part of the consumers.

Price Setter Factors

The price setter is important because of the increased use of dynamic pricing and specifically price-discovery mechanisms. Price-setter transaction characteristics may well affect fairness judgments since attributions of fairness are likely to depend on the level of control the consumer has regarding price determination (Weiner 1985). Fairness heuristic theory provides support for the acceptance of "pricing rules" through the premise that once consumers have established fairness judgments, those judgments serve as heuristics for evaluating new experiences (Van den Bos et al. 1997). If consumers judge a particular pricing rule as fair, subsequent transactions using similar rules will also be perceived as fair. In particular, consumers who participate in price determination through bidding and/or negotiating are more likely to perceive prices as fair. This assertion is consistent with findings that support attribution theory effects associated with locus and controllability (Weiner 1985). When consumers participate in setting the price, the onus is more directly on themselves to ensure price acceptability and they are more likely to make internal attributions regarding the price determination. Therefore, consumers are less likely to perceive a price as unfair, even if evidence exists to the contrary, when they are involved in the price-setting process; the following could be built upon the above assertions (Haws and Bearden, 2006):

1. Consumers will perceive prices determined through bidding as more fair than prices set by the retailer.

2. With price-setter differences, higher prices paid relative to others will trigger weaker negative fairness judgments than seller, time, and/or consumer differences.

Time

As time varies, the expectation of prices remaining constant should be relaxed, although consumers have been shown to insufficiently adjust price perceptions for inflation and often look back at previous prices paid in making fairness judgments (Bolton et al. 2003). Overall, consumers are accustomed to high-low strategies that vary retailers' prices periodically, but they are less accustomed to prices that vary within very short periods of time (e.g., within the same day). Unlike previous research, the current research investigates time differences that are both temporally close and temporally distant. Price discrepancies under close temporal proximity should be more salient and influential than price discrepancies spaced over time. Temporal construal theory suggests that events that are temporally proximal are viewed in more concrete terms, while events that are temporally distant are viewed in more abstract terms (Liberman and Trope 1998). As such, price discrepancies under close temporal proximity should be more salient and influential than price discrepancies spaced over time. Consumers will likely see temporally proximal time-based price changes as unfair. Based on the above,

Haws and Bearden (2006) suggest the following:
Consumers will perceive temporally proximal price differences as more unfair than temporally distant price differences; however, this effect will dissipate over time.

Other consumers

Consumers often have information about the prices paid by others, especially with the easy accessibility of information-sharing mechanisms both online (e.g., chat rooms, blogs, and message boards) and offline (e.g., word of mouth) (Cox 2001). Fairness heuristic research suggests that individuals pay more attention to information about others' outcomes than to other pertinent information (e.g., procedural fairness) when making fairness judgments (Van den Bos et al. 1997). Moreover, social comparisons affect price fairness judgments (Cox 2001).
The concept of equity, which underlies fairness, suggests that generally consumers should pay the same price for the same product (Darke and Dahl 2003). Without explanation, consumers will perceive a price as unfair when it differs from the price paid by other consumers. Even when consumers know that differential prices are based on a buyer's identification (i.e., as a new customer), they regard consumer based price differences as unfair (Grewal et al. 2004). Xia and Monroe (2004) propose that comparisons with other consumers will have a larger effect on perceived price fairness than comparisons with other sellers or self-references.
Based on the above discussions, Haws and Bearden (2006) propose the following:
Higher prices paid relative to other consumers will trigger stronger negative fairness judgments than seller, time, and/or price-setter differences.

An Introduction to Fuzzy Cognitive Maps

Cognitive Maps (CMs) were proposed and applied to ill-structured problems by Axelrod (1976).
The approach used by Axelrod is to develop CM's, i.e. signed digraphs designed to capture the causal assertions of a person, with respect to a certain domain and then use them in order to analyze the effects of alternatives, e.g. policies, business decisions, etc. upon certain goals. A cognitive map has only two basic types of elements Concepts and Causal Beliefs. The concepts are represented as variables and the causal beliefs as relationships among variables. Causal relationships link variables to each other and they can be either positive or negative. Variables that cause a change are called *Cause Variables* while those that undergo the effect of the change in the cause variable are called *Effect Variables*. If the relationship is positive, an increase or decrease in a cause variable causes the effect variable(s) to change in the same direction. If the relationship is negative, then the change which the effect variable undergoes is in the opposite direction. Fig. 1 is a graphical representation of a cognitive map, where variables (X, W, etc.) are represented as nodes, and causal relationships as directed arrows between variables, thus constructing a signed digraph (Nasserzadeh, et al., 2008). A famous example is cited by Vasantha Kandasamy and Smarandache where an expert spells out the five major concepts relating to the unemployed graduated engineers as:
E1 – Frustration
E2 – Unemployment
E3 – Increase of educated criminals
E4 – Underemployment
E5 – Using drugs, etc.

The resulting graph elicited using the expert's opinions and representing variables and corresponding relationships among them is as follows (Figure 2):

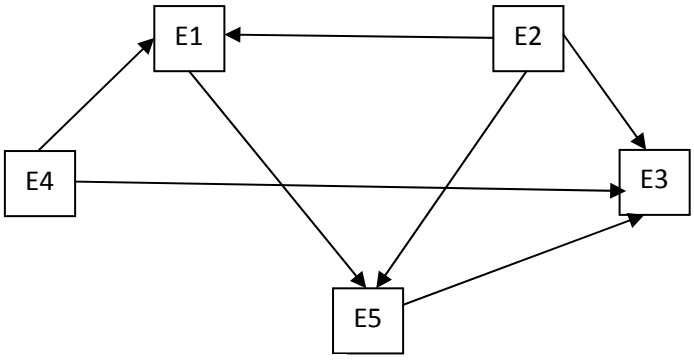


Figure 2-Causal Graph

Cognitive maps were developed in simulation, organizational strategies modeling, support for strategic problem formulation and decision analysis, knowledge bases construction, managerial problems diagnosis, failure modes effects analysis, modeling of social and psychological processes, modeling virtual worlds and analysis of their behavior, requirements analysis and systems requirements specification. (Kardaras and Karakostas, 1999). Kosko (1986) introduces FCM i.e. weighted cognitive maps with fuzzy weights. It is argued, that FCM eliminates the indeterminacy problem of the total effect. Since its development, fuzzy set theory has been advanced and applied in many areas such as expert systems and decision making, control engineering, pattern recognition, etc (Zimmermann, 1991). It is argued that people use fuzzy data, vague rules, etc. and fuzzy sets as a mathematical way to represent vagueness. Fuzzy sets are characterized by a membership function, which is also called the degree or grade of membership.

Different approaches were proposed for the specification of the fuzzy weights in an FCM. One suggestion is to ask the experts to assign a real number from the interval (0, 1) for each relationship and then calculate the average. However, it is difficult for the experts to assign a real number in order to express their beliefs with regard to the strength of relationships. This is the reason why partially ordered linguistic variables such as weak < moderate < strong, etc. are preferred to real number.

It is assumed that a concept in an FCM can be represented by a numerical vector (V), whereas each element (v) of the vector represents a measurement of the concept. Another way of representing a cognitive map is made possible through an adjacency matrix where one can clearly observe the sign of the relationship, while keeping in mind that in case of an absence of relationship between these two factors, the corresponding entry will be empty:

Methodology

For the purpose of determining the strength of each path in the following fuzzy diagram, the mamdani fuzzy operator has been used.

Based upon paths above a questionnaire has been developed which each of the questions refers to one of the defined relationships. These questionnaires have been responded by 30 experts in the fields of pricing and product managers of 4 companies dealing with importing medical equipments. The answers then have been collected and analyzed.

Using the above function, the linguistic labels corresponding to each of the paths shown in figure 5 are quantified. The aim is

to find the path with the greatest impact on the final variable – purchase satisfaction.
Assuming:
C1: seller
C2: consumer
C3: time
C4: price setter
C5: product
C6: terms of sale
C7: price fairness judgment
C8: purchase satisfaction

The paths are as follows:
I_1 = {C1,C7,C8}
I_2 = {C2,C7,C8}
I_3 = {C3,C7,C8}
I_4 = {C4,C7,C8}
I_5 = {C5,C7,C8}
I_6 = {C6,C7,C8}

Results

Based on the mathematics behind fuzzy cognitive maps, the importance weights of each of the paths are computed as follows:

$\Phi^1 = \min(0.74,1) = 0.74$ (strong)
$\Phi^2 = \min(0.72,1) = 0.72$ (strong)
$\Phi^3 = \min(0.42,1) = 0.42$ (medium)
$\Phi^4 = \min(0.61,1) = 0.61$ (medium)
$\Phi^5 = \min(0.65,1) = 0.65$ (medium)
$\Phi^6 = \min(0.52,1) = 0.52$ (medium)
$\Delta = \max \{ \Phi^1, \Phi^2, \Phi^3, \Phi^4, \Phi^5, \Phi^6\}$ = strong

Hence, since the strongest number corresponding to the linguistic label is associated with path 1, this path is the most strategically effective one to affect the consumer's purchase satisfaction.

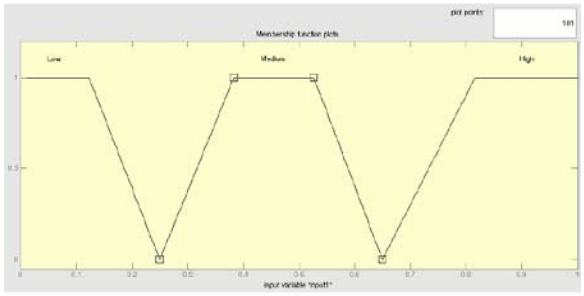


Figure 5-Linguistic Fuzzy Variables

Managerial Implications

- A more precise market targeting

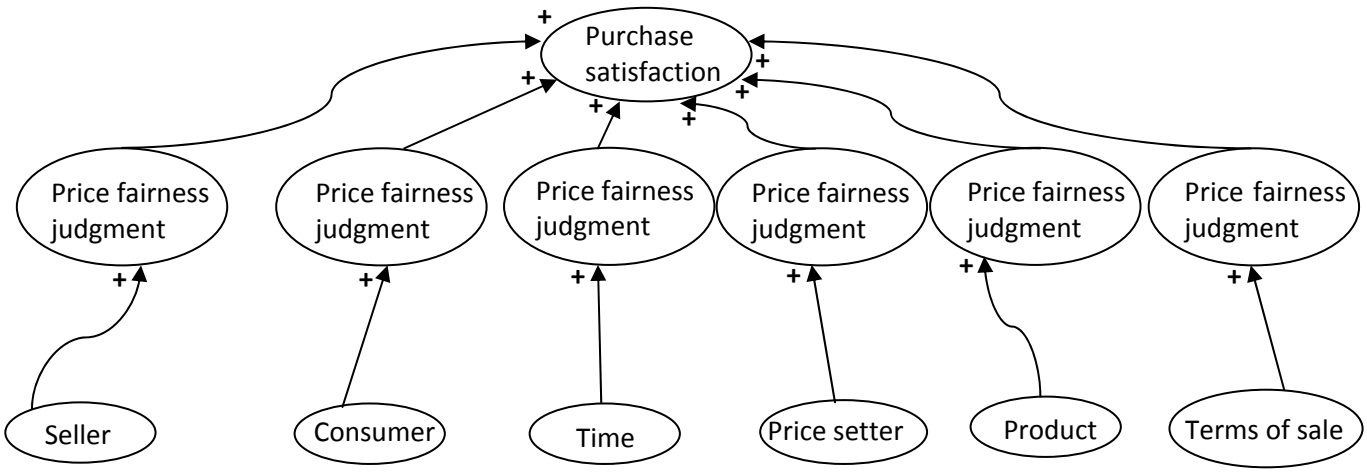


Figure 4-Purchase Satisfaction Cognitive Map

- Objectively defined, updatable promotion scheme
  Since consumers relate more recent price changes to price fairness more than distant price changes, promotions should occur occasionally and from time to time rather than happen in the long run.

- More effective communication of product features
  Consumers more familiar with product attributes will perceive more accurate relationships between attributes and quality thus will have a better implication of the price and its fairness. Accordingly, it is critical to the firm to communicate the so-called attributes and features with the market.

- More educated sales force
  Experienced and knowledgeable sales force will communicate more effectively the product features, the payment terms and lots of other factors which are influential on price fairness judgment and purchase satisfaction.
- Establishing seller-related intangible values for the consumer such as warranties, after-sale services, etc.
  Any factor that is facilitative in purchasing process might influence price fairness judgment through consumers' implication of the transaction; hence it affects consumers' price fairness judgment.

- Working on environmental stimuli having an effect on the consumer's perception process

- Designing the selling process in such a way as to produce seller-related cues in the consumer.

- Seller's effort toward a cultural sacralization (M. R. Solomon, 2008) of the product through seller-related instruments

References:

1. Kelly L. Haws, William O. Bearden, 2006. **Dynamic Pricing and Consumer Fairness Perceptions**, Journal of Consumer Research, Inc. vol. 33
2. Elmaghraby, Wedad and Pinar Keskinocak, 2003. **Dynamic Pricing in the Presence of Inventory Considerations: Research Overview, Current Practices, and Future Directions**, Management Science, 49 (10), 1287–1305.
3. Weiner, Bernard, 1985. "**An Attributional Theory of Achievement Motivation and Emotion**, Psychological Review, 92, 548–73.
4. Van den Bos, Kees, E. Allan Lind, Rie¨l Vermunt, and Henk A. M. Wilke, 1997. **How Do I Judge My Outcome When I Do Not Know the Outcome of Others? The Psychology of the Fair Process Effect**, Journal of Personality and Social Psychology, 72 (5), 1034–46.
5. S. M. Reza Nasserzadeh, M. Hamed Jafarzadeh, Taha Mansouri, Babak Sohrabi, 2008. **Customer Satisfaction Fuzzy Cognitive Map in Banking Industry**, Presented in IBIMA Conference 2008.
6. W. B. Vasantha Kandasamy and Florentin Smarandache, 2003. **Fuzzy Cognitive Maps and Neutrosophic Cognitive Maps**, Xiquan (eBook).
7. Kardaras D, Karakostas B, 1999. **The use of fuzzy cognitive maps to simulate the information systems strategic planning process**, Information and Software Technology 41, pp.197–210.
8. Zimmermann H. J., 1991. **Fuzzy Set Theory and its Applications**, Kluwer Academic Publishers, Dordrecht.
9. Axelrod R. 1976. **Structure of Decision: The Cognitive Maps of Political Elites**, Princeton University Press, Princeton, NJ.
10. Cox, Jennifer L., 2001. **Can Differential Prices Be Fair?**, Journal of Product and Brand Management, 50 (5), 264–75.
11. Darke, Peter R. and Darren W. Dahl, 2003. **Fairness and Discounts: The Subjective Value**

of a Bargain, Journal of Consumer Psychology, 13 (3), 328–38.
12. Lisa E. Bolton, Luk Warlop, Joseph W. Alba, 2003. **Consumer Perceptions of Price (Un)Fairness**, Journal of Consumer Research, Inc., Vol. 29
13. Michael R. Solomon, **Consumer Behavior, Buying, Having, and Being**, Pearson Prentice Hall, 2008 (8th ed.)

A Robust TFPDC Based Power System Stabilizer

H. A. Shayanfar*
E.E.Department
Center of Excellence for Power System Automation and Operation
Iran University of Science and Technology, Tehran, Iran

A. Jalili
Technical Eng. Department
Islamic Azad University- Science and Research Branch, Tehran, Iran

H. Shayeghi
Technical Eng. Department
University of Mohaghegh Ardabili Ardabil, Iran

hashayanfar@yahoo.com, arefjalili@gmail.com, hshayeghi@gmail.com

Abstract- *This paper presents a robust Takagi-Sugeno Fuzzy Parallel Distribution Compensation (TSFPDC) based Power System Stabilizer (PSS) for damping power system low frequency oscillations. The main motivation of using this controller is minimizing the efforts of load disturbances and achieves the desired level of robust performance in the presence of modeling uncertainties. The effectiveness of the proposed method is verified on a New- England test system under various load demand and disturbances in comparison with the Strength Pareto Evolutionary Algorithm (SPEA) based designed PSS through nonlinear time domain simulations and system performance indices. The results evaluation show that the proposed control strategy achieves good robust performance for a wide range of system parameters and load changes in the presence of system nonlinearities and is superior to the other stabilizers.*

Keywords: PSS, Power System Stability, Low Frequency Oscillation.

1 INTRODUCTION

There is no doubt that, in the modern generators, supplementary excitation must be added to mitigate the low frequency oscillations due to negative damping effect of high gain AVRs. Power System Stabilizer (PSS) is used to provide the desired additional damping as a supplementary excitation control [1]. PSS goals as a supplementary excitation, i.e. improving the damping of the low frequency oscillation, power transfer capability and dynamic stability limits in the presence of modeling uncertainties, system nonlinearities and load disturbances, is to determines the power system stability synthesis as a multi objective optimization problem.

In many reported works on the PSS design, several methods such as: classical [2-3], adaptive and variable structure [4-5], robust [6-7] and intelligent methods [8-10] was used for the solution of damping the power system low frequency oscillations. Although, these reported works seem to be good methods for the solution of the PSS synthesis problems. However, the main limitations of these methods are the complexity and time consuming designing process. On the other hand, the investigation carried out using these approaches reveal that it exhibit poor dynamics performance, especially in the presence of the other destabilizing effects such as parameter variations and nonlinearities. In other words, designing process by these methods are more complex and time consuming. Because, the real world power systems are usually large scale systems with complex nonlinear dynamics and variable conditions. Thus, the number of the PSS parameters that should be optimized by highly epistatic objective function (i.e. where parameters being optimized are highly correlated) is large and coordinating among these possible PSSs are difficult. Thus, a flexible controller must be developed to overcome theses drawbacks.

In this paper, a new robust Takagi-Sugeno Fuzzy Parallel Distributed Compensation (TSFPDC) technique based PSS is proposed for damping power system low frequency oscillations. The past few years have witnessed rapidly growing interest in the fuzzy control of nonlinear systems. In particular, the so-called Takagi-Sugeno (T-S) fuzzy model has been widely employed for the control design of nonlinear systems, since it can combine the merits of both fuzzy logic theory and linear system theory [11]. Fuzzy logic theory enables us to utilize qualitative, linguistic information about a complex nonlinear system to decompose the task of the modeling and control design into a group of easier local tasks. At the same time, it also provides the mechanism to blend these local tasks together to yield the overall model and control design [12].

In general, for the simplicity of the practical implementation of the proposed controllers, the PI controller with slip available at the location of each controlled device is most favorable. In order that this newly proposed controller perform well, the gains of the PI controller based PSS of each subsystem must be carefully designed. Particle Swarm Optimization (PSO) technique is used for the optimal tuning of the PI gains in each subsystem to improve the optimization synthesis and the speed of the algorithms convergence.

The proposed controller is tested on a 10 machine 39 bus power system under different operation conditions in comparison with the Strength Pareto Evolutionary Algorithm (SPEA) based deigned PSS (see Ref. [13] for more details about the problem solution) controller through Integral of the Time multiplied Absolute value

* Corresponding Author. E-Mail Address: hashayanfar@yahoo.com (H. A. Shayanfar)

of the Error (ITAE) and Figure of Demerit (FD) performance indices. To illustrate the effectiveness of the proposed method, two scenarios under large load demands have been simulated. Results evaluation show that the proposed PSS has an excellent capability in damping power system low frequency oscillations and enhances greatly the dynamic stability of the power systems. Also, the newly proposed strategy achieves good robust performance for a wide range of load changes in the presence of system nonlinearities and is superior to the other PSSs.

2 Power system description

The complex nonlinear model related to an *n*-machine interconnected power system, can be described by a set of differential-algebraic equations by assembling the models for each generator, load, and other devices such as controls in the system, and connecting them appropriately via the network algebraic equations. In this study, the two-axis model is used for the time domain simulations. For a given operating condition, the multi-machine power system is linearized around the operating point for synthesis procedure. Nonlinear model of a multi machine power system is described by a set of equation as given by [14]:

$$\dot{\omega}_i = \frac{T_{mi} - T_{ei}}{2H} \tag{1}$$

$$\dot{\delta}_i = \omega_0(\omega_i - 1) \tag{2}$$

$$\dot{E}'_{qi} = \frac{E_{fdi} - (E'_{qi} + (x_{di} - x'_{di})I_{di})}{T'_{doi}} \tag{3}$$

$$\dot{E}_{fdi} = \frac{K_{Ai}(V_{ref} - V_{ti}) + u_{si}) - E_{fdi}}{T_{Ai}} \tag{4}$$

$$T_e = E'_{di}I_{di} + E'_{qi}I_{qi} - (x'_{qi} - x'_{di})I_{di}I_{qi} \tag{5}$$

$$E = E'_{qi}I_{di} - (x_{di} - x'_{di})I_{di} \tag{6}$$

3 Proposed control design scheme

3.1 The T-S fuzzy model

In this stage, the design procedure begins with representing a given nonlinear plant by the so-called Takagi-Sugeno (T-S) fuzzy model. The fuzzy model proposed by Takagi and Sugeno is described by fuzzy IF-THEN rules which represents local linear input-output relations of a nonlinear system [15-17]. The main feature of a T-S fuzzy model is to express the local dynamics of each fuzzy implication (rule) by a linear system model. The overall fuzzy model of the system is achieved by the fuzzy ''blending'' of the linear system models. The i^{th} rules of the T-S fuzzy models are of the following forms [16]:

Model Rule i:

If $Z_1(t)$ *is* M_{i1} *... and* $Z_p(t)$ *is* M_{ip}

$$\text{Then} \quad \begin{cases} \dot{x}(t) = A_i x(t) + B_i \\ y(t) = C_i x(t) \end{cases} \quad i = 1,2,...,r \tag{7}$$

Where, M_{ij} is the fuzzy set and r is the number of model rules; $x(t) \in R^n$ is the state vector, $u(t) \in R^m$ is the input vector, $y(t) \in R^q$ is the output vector, $A_i \in R^{n\times n}$, $B_i \in R^{n\times m}$, and $C_i \in R^{q\times n}$; $z_1(t),..., z_p(t)$ are known premise variables that may be functions of the state variables and external disturbances. Each linear consequent equation represented by $A_i x(t)+B_i u(t)$ is called a ''subsystem''.

Given a pair of $(x(t), u(t))$, the final outputs of the fuzzy systems are inferred as:

$$\dot{x}(t) = \frac{\sum_{i=1}^{r} \omega_i(z(t))\{A_i x(t) + B_i u(t)\}}{\sum_{i=1}^{r} \omega_i(z(t))} = \sum_{i=1}^{r} h_i(z(t))\{A_i x(t) + B_i u(t)\} \tag{8}$$

$$y(t) = \frac{\sum_{i=1}^{r} \omega_i(z(t))C_i x(t)}{\sum_{i=1}^{r} \omega_i(z(t))} = \sum_{i=1}^{r} h_i(z(t))C_i x(t)$$

3.2. Parallel distribution compensation

The Parallel Distributed Compensation (PDC) offers a procedure to design a fuzzy controller from a given T-S fuzzy model. To realize the PDC, a controlled object (nonlinear system) is first represented by a T-S fuzzy model. In the PDC design, each control rule is designed from the corresponding rule of a T-S fuzzy model. The constructing following fuzzy controller via the PDC is given by [16]:

Control Rule i:

If $Z_1(t)$ *is* M_{i1} *... and* $Z_p(t)$ *is* M_{ip}

$$\text{Then} \quad u(t) = -F_i x(t) \quad i = 1,2,...,r \tag{9}$$

The fuzzy control rules have a linear controller (PI control laws in this case) in the consequent parts. The overall fuzzy controller is represented by:

$$u(t) = -\frac{\sum_{i=1}^{r} \omega_i(z(t))F_i x(t)}{\sum_{i=1}^{r} \omega_i(z(t))} = -\sum_{i=1}^{r} h_i(z(t))F_i x(t) \tag{10}$$

The fuzzy controller design is to determine the gains of PI controller ($F_i=K_{pi}+K_{ii}/s$) in the consequent parts. With PDC, we have a simple and natural procedure to handle the nonlinear control systems. Other nonlinear control techniques require special and rather involved knowledge.

3.3. Decentralized TSFPDC design

The structure of the power system component is shown in Fig. 1 to obtain the complete system model for carrying out the transient stability studies. In this system, the excitation power is supplied through a transformer from the generator terminals. High gain exciters may add negative damping to the system and worsen its relative stability. To overcome that, supplementary excitation must be added and for this purpose the conventional lead-lag PSS is widely used. In some papers the function of a PSS is to add the damping to the generator rotor oscillations. This is

achieved by modulating the generator excitation so as to develop a component of electrical torque in phase with the rotor speed deviations. Such a way of producing the damping torque is the most cost-effective method of enhancing the small signal stability of the power system, when fast acting high gain excitation systems are used. Thus, this paper presents a robust decentralized Power System Stabilizer (PSS) based on Takagi-Sugeno Fuzzy Parallel Distribution Compensation (FPDC) technique for damping of the power system low frequency oscillations.

Power systems consist mainly of a set of generating units, a transmission network and loads. These units interact with each other through active and reactive power generation (P,Q) over the transmission network. Briefly, any power system is composed of a set of inherently interacting subsystems, where each subsystem consists of a generating-unit connected to the rest of the system by a tie line whose reactance is the equivalent Thevenin's reactance at the terminal bus (X_e=X_{Th}). For designing the proposed TSFPDC based PSS in this study, a subsystem is considerably approximated by a single-machine connected to an infinite system. This assumption is made possible because fuzzy modeling allows imprecision. As a result of this approximation, each generator can be decoupled from the entire system. The influence of the rest of the system will be taken care by the scheduling variables; namely its real and reactive powers (P,Q) and an equivalent tie line reactance (X_e). All possible dynamics at the interface between a generator and the rest of the system are assumed to be reflected by this set of scheduling variables (P, Q, X_e). This decoupling leads to a decentralized control scheme for the synthesis procedures.

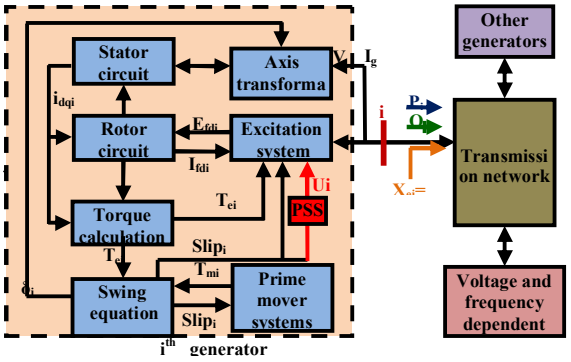


Fig. 1. The structure of the complete power system

The origin of the power systems uncertainties are the continuous variations in the load patterns and transmission network. Since the system is to be linearized around the equilibrium point, it follows that a different system triple (A, B, C) is obtained for each operating point. It is assumed that the set of variables (P, Q, X_e) of certain subsystem varies independently over the following ranges: $P \in [P_- \ P_+]$, $Q \in [Q_- \ Q_+]$ and $X_e \in [X_{e-} \ X_{e+}]$. These ranges are selected to encompass all practical operating points and very weak to very strong transmission networks. Possible combinations of minimum and maximum values of these variables result in eight operating points corresponding to the vertices of a cuboids in the (P, Q, X_e) space. Consequently, a set of matrices obtained from an operating point can be represented by $(A, B, C) \in \Omega$, where:

$$\Omega = \left\{ (A,B,C) : (A,B,C) = \sum_{i=1}^{8} \alpha_i (A_i, B_i, C_i), \alpha_i \geq 0 \right\} \quad (11)$$

The set Ω describes a polytope with eight vertices $(A_i, B_i, C_i), i = 1, 2,...,8$ calculated at $[P_- \ Q_- \ X_{e-}]$, $[P_- \ Q_- \ X_{e+}]$,..., $[P_+ \ Q_+ \ X_{e+}]$, respectively.

Changes in load and system topology or most of the system parameters lead to uncertainties in the state-matrix A. Uncertainties in the input matrix B can only is caused by the parametric variations in the excitation system and are not taken into account in this study. Rotor speed deviation is selected as the measured output and thus, no uncertainties are appeared in matrix C.

Each vertex system in the polytope (11) corresponds to a model rule in a T-S fuzzy system which is stated as:

Model Rule 1:

If $(P$ *is* $P_-)$ *and* $(Q$ *is* $Q_-)$ *and* $(X_e$ *is* $X_{e-})$

Then $\begin{bmatrix} \dot{x} \\ y \end{bmatrix} = \begin{bmatrix} A_1 & B \\ C & o \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}$

Model Rule 2:

If $(P$ *is* $P_-)$ *and* $(Q$ *is* $Q_-)$ *and* $(X_e$ *is* $X_{e+})$

Then $\begin{bmatrix} \dot{x} \\ y \end{bmatrix} = \begin{bmatrix} A_2 & B \\ C & o \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}$

…,

Model Rule 8:

If $(P$ *is* $P_+)$ *and* $(Q$ *is* $Q_+)$ *and* $(X_e$ *is* $X_{e+})$

Then $\begin{bmatrix} \dot{x} \\ y \end{bmatrix} = \begin{bmatrix} A_8 & B \\ C & o \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}$

The resulting fuzzy system is inferred as the weighted average of the local models and is given as:

$$\begin{bmatrix} \dot{x} \\ y \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{8} (\omega_i A_i) & B \\ C & o \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \quad (12)$$

Any value $P \in [P_- \ P_+]$ can be expressed as $P = L_1(P_-, P_+, P) \times P_- + L_2(P_-, P_+, P) \times P_+$. Where, $L_1(P_-, P_+, P)$ *and* $L_2(P_-, P_+, P)$ are the membership functions for the variable P such that $L_1(P_-, P_+, P) + L_2(P_-, P_+, P) = 1$, and consequently these membership functions can be calculated as:

$$L_1(P_-, P_+, P) = \frac{P_+ - P}{P_+ - P_-}, L_2(P_-, P_+, P) = \frac{P - P_-}{P_+ - P_-} \quad (13)$$

The membership functions $L_1(P_-, P_+, P)$ and $L_2(P_-, P_+, P)$ are labeled "L_1" and "L_2", respectively. Figure 2-a shows the membership functions for the variable P.

Similarly, the membership functions for Q and X_e are defined and labeled M_1, M_2 and N_1, N_2 , respectively. Figures 2-b and 2-c show the membership functions for the variable Q and X_e, respectively. The weights are calculated as $h_1 = L_1M_1N_1$, $h_2 = L_1M_1N_2$, $h_3 = L_1M_2N_1$,..., and $h_8 = L_2M_2N_2$.

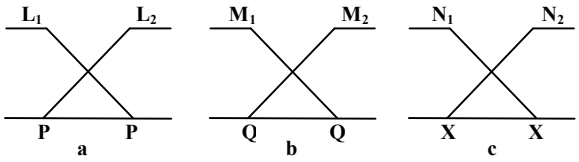


Fig. 2. Membership functions for scheduling: a) Variable P; b) Variable Q; c) Variable X_e

Typically, in a PSS, the speed deviation of the generator is used as a control signal. In such case, attention is oriented towards the PI controller design methods. Thus, the design of the PI controller based PSS for power systems is described by continuous proposed models. A fuzzy based output-feedback PSS shares the same fuzzy sets with the fuzzy model as follows:

$$u(t) = \sum_{i=1}^{8} \omega_i F_i y(t) = \sum_{i=1}^{8} \omega_i \omega_j \{F_i C_j x(t)\} \quad (14)$$

Where, F_i are ($F_i = K_{pi} + K_{ii}/s$) that the gains of F_i to be determined. By substituting Eq. (15) in T-S model (9), we have:

$$\dot{x}(t) = \sum_{i=1}^{8}\sum_{j=1}^{8}\sum_{l=1}^{8} \omega_i \omega_j \omega_l \{A_i + B_i F_j C_l\} x(t) \quad (15)$$

For the case of power systems, $B_i = B$, $C_l = C$, $i, l = 1,2,\ldots, r$. Then Eq. (16) can be rewritten as:

$$\dot{x}(t) = \sum_{i=1}^{8} \omega_i \{A_i + BF_iC\} x(t) \quad (16)$$

Fig. 3 show the structure of the proposed controller as PSS and the flowchart of the design steps using the proposed method is shown in Fig. 4.

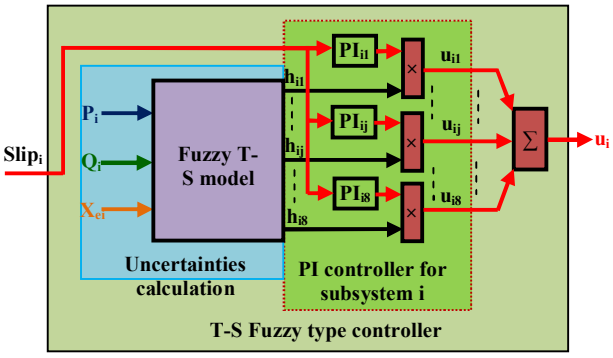


Fig. 3. Schematic diagram of the proposed PSS

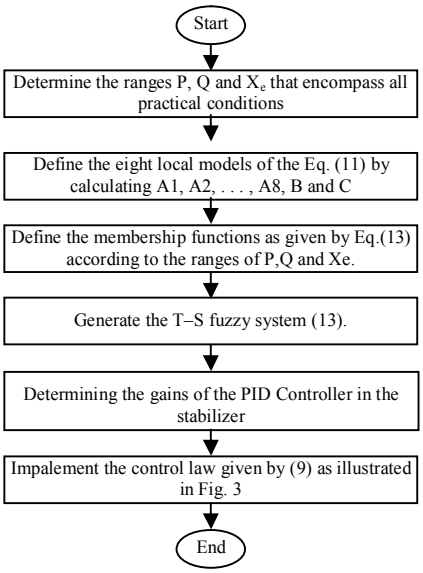


Fig. 4. Flowchart of the design steps

In order for a proposed fuzzy based control system to perform well, the gains of the PI controller in each subsystem must be carefully designed. For this reason, Particle Swarm Optimization (PSO) technique is used for the design of the PI controller gains of each subsystem stabilizer in an operation condition which have the features of easy implementation, short execution time and robust mechanisms of escaping from the local optimum. Also, is a promising tool for the engineering applications (see Ref. [18]). It should be noted that the choice of the properly fitness function is very important in the synthesis procedure. Because, different fitness functions promotes different PSO behaviors. For our optimization problem, an ITAE performance index is taken as the objective function as follows:

$$ITAE = 10 \times \int_0^{10} t|\Delta\omega_i|dt \quad (17)$$

4. Case study

In this study, the New England 10-machine 39 bus power systems shown in Fig. 5 is considered as a test system. Detail of the system data are given in Ref. [8]. It is assumed that the set of variables (P,Q,X_e) of certain subsystem varies independently over the following ranges: $P \in [3 \;\; 12]$, $Q \in [-1 \;\; 4]$ and $X_e \in [0.04 \;\; 0.06]$. The eight local models of the Eq. (11) are defined according to the eight loading condition. In order to acquire better performance, number of particle, particle size, number of iteration, c_1 and c_2 are chosen as: 20, 1, 500, 2 and 2, respectively. Also, the inertia weight, w, is linearly decreasing from 0.9 to 0.4. Because of the stochastic nature of the proposed PSO, four trials are performed with 100 iterations and the best solution amongst them is considered as the final solution. The near optimal solution is achieved after approximately 40

iterations. The results show that the PSO consistently converged to the optimal solution. The optimal PID controller gains in the proposed method for the different operation conditions as given in Table 1 using the multi objective function of Eq. (18) are listed in Table 1.

5. Nonlinear time-domain simulation

To assess the effectiveness and robustness of the proposed method, three different cases designated as nominal (data given in ref [8])), lightly (25% decreased from the nominal value) and heavily (25% increased from the nominal value) loading conditions are considered. The effectiveness of the proposed TSFPDC based PSS is tested on the multi machine power system (as shown in Fig. 5) under different operation conditions in comparison with the SPEA based designed [13] through the nonlinear time simulation and some performance indices.

5.1. Scenario 1

In this scenario, the heavily loading condition is considered and assume that a 0.5 *p.u.* step increasing in the mechanical torque of each generator was applied at t=1 s. The system response to this disturbance is shown in Fig. 6. It can be seen that the proposed method has good damping characteristics for low frequency oscillations and stabilizes the system quickly.

5.2. Scenario 2

In this scenario, the lightly loading condition is considered to analysis the performance of the proposed controller under transient conditions. This scenario is verified by applying a three-phase fault at t=1 sec, on bus 25 at the end of the line 26-25. The fault is cleared by permanent tripping of the faulted line at t=1.05 sec. The speed deviations of all generators are shown in Fig. 7. Using the proposed method, the speed deviation of all generators are quickly driven back to zero and has small settling time. Moreover, it can be seen that the proposed TSFPDC based PSSs achieves good robust performance and provides superior damping in comparison with the SPEA based designed PSS.

Table 1: Optimal PID gains for the proposed method

Gain		Gen. 1	Gen. 2	Gen. 3	Gen. 4	Gen. 5	Gen. 6	Gen. 7	Gen. 8	Gen. 9	Gen. 10
K_p	Case 1	479.6	489.8	513.1	515.2	506.4	495.5	491.5	478.3	476.0	538.9
	Case 2	505.3	495.9	475.1	514.5	499.9	510.6	494.7	488.5	511.0	502.9
	Case 3	524.1	528.4	546.6	508.9	517.8	518.6	497.4	528.3	521.0	561.8
	Case 4	535.7	558.4	570.9	542.8	533.9	563.9	523.8	548.6	543.5	569.5
	Case 5	216.8	222.1	229.8	226.1	222.2	224.5	216.4	215.7	213.1	220.1
	Case 6	191.7	192.6	191.0	194.4	187.3	189.0	192.0	195.4	188.3	207.4
	Case 7	99.2	96.5	105.8	101.5	97.5	102.5	97.67	99.2	102.3	110.3
	Case 8	118.0	113.0	118.2	121.7	112.5	122.0	116.2	111.4	115.4	123.0
K_i	Case 1	4331.1	4366.4	4160.0	4513.9	4302.2	4204.9	4141.9	4387.8	4425.5	4683.3
	Case 2	4275.3	4093.3	4215.7	4419.8	4381.6	4329.7	4335.3	4170.8	4305.3	4397.1
	Case 3	2064.0	2045.3	2174.3	2136.0	2192.9	2174.5	2035.57	2182.1	2097.7	2251.7
	Case 4	2176.9	2155.0	2166.3	2188.4	2104.7	2084.3	2022.3	2086.1	2017.1	2140.6
	Case 5	1076.5	1049.4	1001.1	1039.8	995.6	1095.6	1024.7	1023.1	1012.0	1069.9
	Case 6	1033.5	1006.9	1052.1	1021.6	1036.4	1085.3	1002.5	1079.8	1059.5	936.9
	Case 7	270.1	274.4	260.0	262.8	270.8	262.3	255.4	273.5	264.9	253.5
	Case 8	264.9	242.0	268.0	252.7	264.7	261.6	238.8	249.2	267.5	250.1

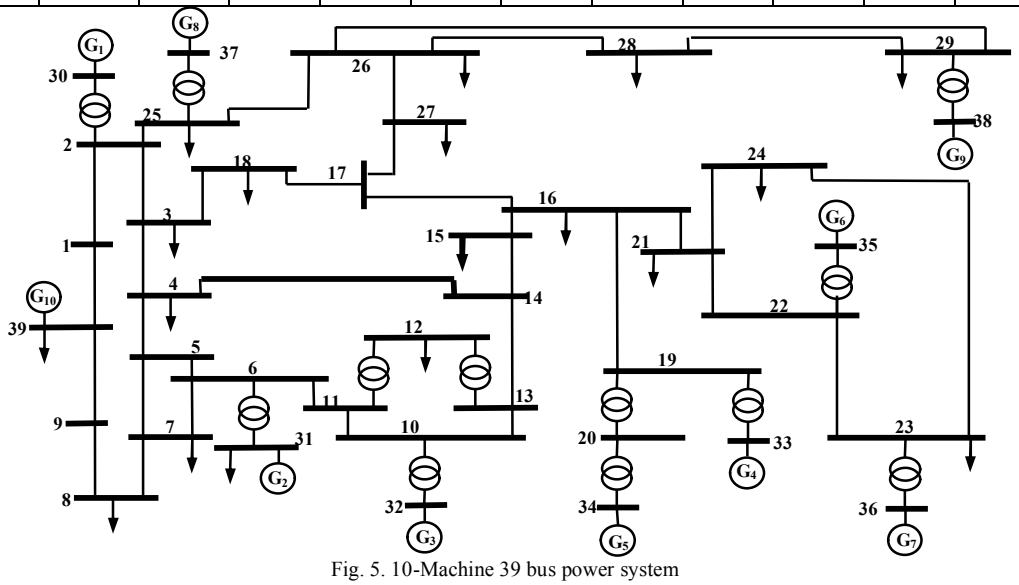


Fig. 5. 10-Machine 39 bus power system

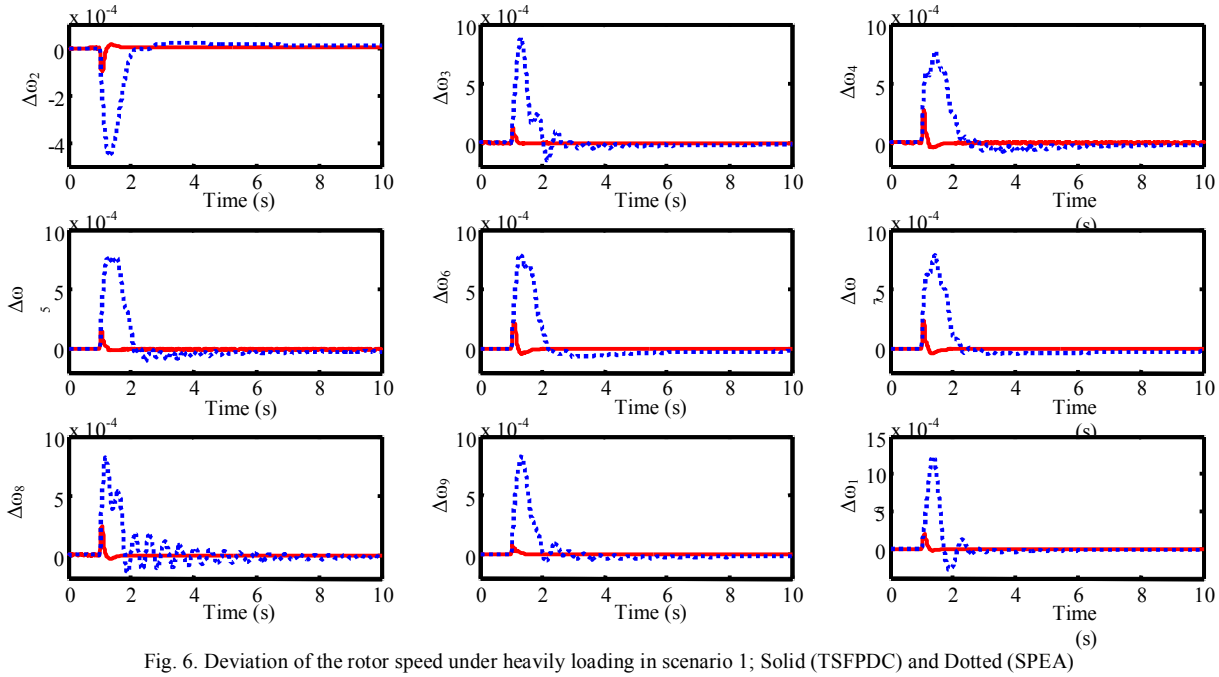


Fig. 6. Deviation of the rotor speed under heavily loading in scenario 1; Solid (TSFPDC) and Dotted (SPEA)

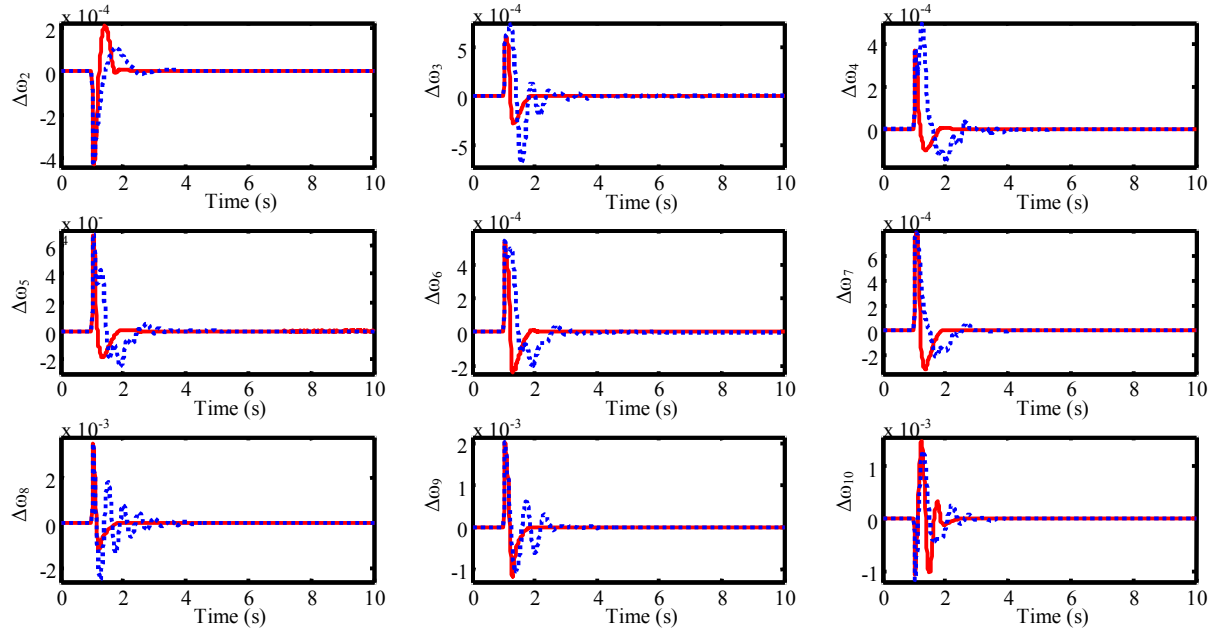


Fig. 7. Deviation of the rotor speed in Scenario 2 under lightly loading, Solid (TSFPDC) and Dotted (SPEA)

To demonstrate the performance robustness of the proposed method, the ITAE and FD performance indices based on the system performance characteristics are being used as:

$$ITAE = 10^4 \times \int_0^{10} t \sum_{i=1}^{10} |\Delta\omega_i| dt \quad (19)$$

$$FD = \frac{\sum_{i=1}^{10} [(OS_i \times 10^3)^2 + (US_i \times 10^3)^2 + (Ts_i)^2 \times 10]}{10} \quad (20)$$

Where, Overshoot (*OS*) Undershoot (*US*) and settling time (for 5% band) of the speed deviation of generator *i* is considered for the evaluation of the FD. It is worth mentioning that the lower the value of these indices is, the better the system response in terms of the time-domain characteristics. Numerical results of the performance robustness for all loading conditions are shown in Table 2; whereas the load of power system are varied from -25% to 25% of the nominal values. It can be seen that the values of these system performance characteristics with the proposed PSSs are much smaller in comparison with the SPEA designed approach. This

demonstrates that the overshoot, undershoot, settling time and speed deviations of all machines are greatly reduced by applying the proposed T-S fuzzy PDC typed PSSs.

Table 2. ITAE and FD values

Load changes	ITAE			
	TSFPDC	SPEA	TSFPDC	SPEA
-25%	1.24	25.28	143.63	616.23
-20%	1.18	25.31	133.92	600.84
-15%	1.13	25.35	126.63	599.36
-10%	1.09	25.42	117.61	604.31
-5%	1.06	25.5	113.79	679.31
Nominal	1.05	25.61	111.05	684.82
5%	1.03	25.75	110.13	896.61
10%	1.03	25.93	108.99	936.69
15%	1.02	26.17	108.58	1058.08
20%	1.03	26.5	108.56	1182.6
25%	1.04	27.11	109.1	1353.84

6. Conclusions

A new robust PSS for damping the power system low frequency oscillations using the Takagi-Sugeno fuzzy parallel distribution compensation is proposed in this paper. It has been shown that using the T-S FPDC technique, each local subsystem PSS can be designed independently such that the stability of the overall closed loop system is achieved in the presence of wide range load variation and nonlinearities. Since, each subsystem in the interconnected power system contains different kinds of uncertainties and disturbances. Thus, power system stabilizer design problem has been formulated as a multi objective optimization control problem via TSFPDC control approach and solved by the PSO technique to obtain the optimal gains of the PI controllers. The effectiveness of the proposed strategy was tested on the New England 10 machine 39 bus power systems and compared with the SPEA based designed PSS under load chances and disturbances through ITAE and FD performance indices. The simulation results show that the proposed PSS achieves good robust performance such as damping power system low frequency oscillations for wide range of load change in the presence of system nonlinearity and disturbances and is superior to other controllers.

References

[1] P. Kundur, "Power system stability and control", McGraw-Hill Inc., New York, 1994.

[2] Y. Yu, Electric power system dynamics. Academic Press, 1983.

[3] Y. Hsu, C. Hsu, "Design of proportional integral power system stabilizer," IEEE Trans Transaction on Power Systems, PWRS, Vol. 1, No. 2, pp. 46-53, 1986.

[4] D. Pierre, "A perspective on adaptive control of power systems," IEEE Transaction on Power Systems, PWRS, Vol. 2, No. 2, pp. 387-396, 1987.

[5] S. Cheng, O. Malik, G. Hope, "Damping of multi-modal oscillations in power systems using a dual-rate adaptive stabilizer" IEEE Transaction on Power Systems, Vol. 3, pp. 101-108, 1988.

[6] A. Hardiansyah, S. Furuya, J. Irisawa , "A robust H_∞ power system stabilizer design using reduced-order models," Electric Power Energy Systems, Vol. 28, pp. 21-28, 2006.

[7] A. Abdel Ghany, "Design of a mixed H_2/H_∞ robust PID power system stabilizer with fuzzy adaptation and simulated annealing optimization," IEEE Transactions on Energy Conversion, pp. 316- 324, 2008.

[8] D. Chatuvedi, O. Malik, "Neurofuzzy power system stabilizer," IEEE Transactions on Energy Conversion, Vol. 23, No. 3, pp. 887-894, 2008.

[9] A. El-zonkoly, A. Kalil and Ahmied N, "Optimal tuning of lead lag fuzzy logic power system stabilizer using particle swarm optimization," Expert System with Applications, Vol. 10, pp. 1-10, 2008.

[10] H. Shayeghi, H. A. Shayanfar, A. Safari, R. Aghmasheh, "A robust PSSs design using PSO in a multi machin environment," Energy Conversion and Management, Vol. 51, pp. 696- 702, 2010.

[11] M. Soliman, A. Elshafei, F. Bendary and W. Mansour, "LMI static output feedback design fuzzy power system," Expert Systems with Applications, Vol. 36, pp. 6817-6825, 2009.

[12] S. Ghoshal, A. Chatterje, V. Mukherjee, "Bio-inspired fuzzy logic based tuning of power system stabilizer," Expert Systems with Applications, Vol 36, pp. 9281-9292, 2009.

[13] H. Yassami, A. Darabi and S. Rafiei, "Power system stabilizer design using Strength Pareto multi-objective optimization approach," Electric Power System Research, Vol. 80, pp. 838-846, 2010.

[14] K. Padyar, "Power system dynamics stability and control," BS Publication, Second Edition 2006.

[15] A. Jalili, H. Shayeghi and H. A. Shayanfar, "T-S fuzzy parallel distribution compensation controller for power system stabilizer," 5th International Conference on Technical and Physical Problems of Power Engineering Spain, pp. 180-174, 2009.

[16] K. Tanaka, H.Wang, "Fuzzy control systems design" and analysis: A linear matrix inequality approach, New York: Wiley 2001.

[17] T. Guerra and L. Vermeiren, "LMI-based relaxed nonquadratic stabilization conditions for nonlinear systems in the Takagi-Sugeno's form," Automatica, Vol. 40, No. 5, pp. 823-829, 2004.

[18] H. Shayeghi H, A. Pirayeshnegab, A. Jalili and H.A. Shayanfar, "Application of PSO technique for GEP in restructured power systems," Energy Conversion and Management, Vol. 50, pp. 2127-2135, 2009.

GSA to Tune Fuzzy Controller for Damping Power System Oscillation

H. A. Shayanfar *
Electrical Eng. Department, Islamic Azad University,
South Tehran Branch, Tehran, Iran
E-mail: hashayanfar@yahoo.com

O. Abedinia
Islamic Azad University, Young Researcher Club,
Ardabil Branch, Ardabil, Iran
E-mail: oveis.abedinia@hotmail.com

Mohammad S. Naderi
School of Electrical Eng. & Telecommunication,
UNSW, Sydney, Australia
E-mail: m.salaynaderi@unsw.edu.au

A. Ghasemi
Technical Eng. Department, University of
Mohaghegh Ardabili, Ardabil, Iran
E-mail: ghasemi.agm@gmail.com

Abstract— **This paper presents a Gravitational Search Algorithm (GSA) to tune optimal rule-base of a Fuzzy Power System Stabilizer (FPSS) which leads to damp low frequency oscillation following disturbances in power systems. Usually in a rule based fuzzy control system, selection of suitable rules is more difficult, because of its complexity. Thus, to reduce the design effort and find a better fuzzy system control, the optimal decision making rules is constructed using the GSA which leads to design controller with simple structure that is easy to implement. The New England 10-machine 39-bus standard power system in comparison with the Tabu Search (TS) PSS, under various system configurations and loading conditions, is employed to illustrate the performance of the proposed method.**

Keywords: FPID, PSS, Fuzzy Rule-base, GSA, Multi-machine Systems.

I. INTRODUCTION

The dynamic stability of power systems is an important factor for secure system operation. Low-frequency oscillation modes have been observed when power systems are interconnected by weak tie lines [1]. The low-frequency oscillation mode, which has poor damping in a power system, is also called the electromechanical oscillation mode and usually occurs in the frequency range of 0.1–2.0 Hz [2]. The Power System Stabilizer (PSS) has been widely used for mitigating the effects of low-frequency oscillation modes. The construction and parameters of PSS have been discussed in many papers [3]. Currently, many plants prefer to employ conventional lead-lag structure PSSs, due to the ease of online tuning and reliability.

Genetic Algorithm (GA) is a powerful optimization technique, independent on the complexity of problems where no prior knowledge is available. Many PSS tuning methods using GA were presented in [4-5]. These works investigated the use of genetic algorithms for simultaneously stabilization of multi-machine power system over a wide range of scenarios via power system stabilizers with fixed parameters. Although GA is very sufficient in finding global or near global optimal solution of the problem, it requires a very long run time that may be several minutes or even several hours depending on the size of the system under study [6].

Recently, Fuzzy based PSS (FPSS) schemes have been proposed [7-8]. Fuzzy systems work with a set of linguistic rules, which are carefully designed by experienced operators. It is a model-free approach, which is generally considered suitable for controlling imprecisely defined systems. In fuzzy control, the controller is synthesized from a collection of fuzzy If–Then rules which describe the behavior of the unknown plant. Accordingly, selection of an appropriate rule set is more difficult because it is a computationally expensive combinatorial optimization problem. Sometimes for fuzzy controllers, rule base is derived from human expert who have acquired their knowledge through experience [9]. However, experts may not always be available even when available extraction of an appropriate set of rules from the expert may be tedious, time consuming and process specific. Thus, extraction of an appropriate set of rules or selection of an optimal set of rules from the set of possible rules is an important and essential step toward the design of any successful fuzzy logic controllers.

Consequently, in this paper, in order to overcome these problems, a Gravitational Search Algorithm (GSA) based PID type controller is proposed for the solution of the PSS problem. Here, the GSA optimization algorithm is used for the optimal tuning of the fuzzy rule-base to improve the optimization synthesis and damping low frequency oscillations following disturbances in power systems. In this study, the fuzzy PID controller is constructed as a set of control rules and then gains of PID

* Correspanding Author. E-Mail Address: hashayanfar@yahoo.com (H. A. Shayanfar)

controller are tuned online from the knowledge base and the fuzzy inference. To reduce the design effort, the fuzzy system is designed automatically by GSA. GSA is constructed based on the law of Gravity and the notion of mass interactions. The GSA algorithm uses the theory of Newtonian physics and its searcher agents are the collection of masses. Unlike many evolutionary algorithms such as GA and Neural Networks, the GSA will not get trapped in local optimum [10].

The effectiveness of the proposed FPSS is tested on a multi-machine power system under different operating conditions in comparison with the Tabu Search (TS) based PSS (TSPSS) [11] through nonlinear time simulation and some performance indices. Results evaluation show that the proposed method achieves good robust performance for damping low frequency oscillations under different operating conditions and is superior to the other methods.

II. STRUCTURE OF FUZZY PID CONTROLLER

Recently the research for control methods based on Fuzzy Logic Controllers (FLC) as PSS has greatly improved the dynamic characteristics of power system [7]. Because of the complexity and multi-variable conditions of the power system, conventional control methods may not give satisfactory solutions. On the other hand, their robustness and reliability make fuzzy controllers useful for solving a wide range of control problems in power systems. In general, the application of the fuzzy logic to PID control design for the PSS design can be classified in two major categories according to the way of their construction [9]:

- A typical PSS is constructed as a set of heuristic control rules, and the control signal is directly deduced from the knowledge base.
- The gains of the conventional PID controller are tuned on-line in terms of the knowledge based and fuzzy inference, and then, the conventional PID controller generates the control signal.

Fig. 1 shows the block diagram of the classical fuzzy type controller to PSS design for each generator.

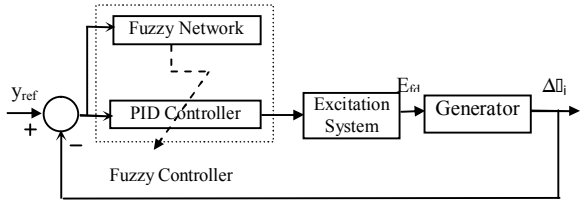


Figure 1. The classical FPID controller design

In the design of fuzzy logic controller, there are five parts of the fuzzy inference process:

- Fuzzification of the input variables.
- Application of the fuzzy operator (AND or OR) in the anteceden.
- Implication from the antecedent to the consequent.
- Aggregation of the consequents across the rules.
- Defuzzification.

The controller block of fuzzy is formed by fuzzification of ($\Delta\omega_i$), the interface mechanism and defuzzification. Therefore, u_i is a control signal that applies to the excitation system in each generator. By taking $\Delta\omega_i$ as the system output, control vector for the conventional PID controller is given by:

$$u_i = K_{Pi}\Delta\omega_i(t) + K_{Ii}\int_0^t \Delta\omega_i(t)dt + K_{di}\Delta\dot{\omega}_i(t)$$

The parameters, K_{Ii}, K_{di} and K_{pi} are determined by a set of fuzzy rules of the form:

If $\Delta\omega_i$ is A_i and $\Delta(\Delta\omega_i)$ is B_i then, K_{di} is C_i and K_{pi} is D_i and K_{Ii} is E_i, $i=1,2,\cdots,n$.

Where, A_i, B_i, C_i, D_i and E_i are fuzzy sets on the corresponding supporting sets.

In many cases, the performance of FPID controller depends on a designed knowledge base in which fuzzy control rules are defined [12]. In the traditional method, the rule base is determined by experience and control knowledge of human expert. However, it is a trial and error process and takes much time and cost. Thus, the automatic design method for FPID control, which can generate an optimal rule tables without human experts, is desirable. Here, we make use of GSA technique to find optimal rule sets of the FPID based PSS.

III. GSA TO DESIGN FUZZY CONTROLLER

GSA is a new search algorithm that has been proven efficient in solving many problems. One of the important advantages of the GSA is its ability to find the global optimum in a shorter period of time in comparison with previous optimization algorithms.

This paper, proposed GSA-based Fuzzy (GSAF) controller for the PSS design, which combines the advantage of the GSAs and fuzzy control techniques to achieve good robust performance. GSA has a great potential in solving complex power system problems. The motivation of using the proposed GSAF PSS is to reduce the fuzzy system effort and determine the optimal controller scheme such that the relative stability is guaranteed and the time domain specifications concurrently secured. There is no doubt that, obtaining the optimal decision-making logic for the proposed GSAF control strategy is very important to achieve the desired level of system robust performance, because it is a computationally expensive combinatorial optimization problem. Usually, the rule base sets are determined by experience and control knowledge of human expert. But,

expert may not always be available even where available it is a trial-error process and takes much time and cost [9]. Hence, to overcome these drawbacks and reduce fuzzy system effort and cost, a modified GSA technique is being used to construct the optimal rule base sets of the proposed fuzzy type PID PSS. Fig. 2 shows the structure of the proposed GSAF PSS to improve power system stability.

In the proposed rule base optimization problem, the membership function sets for the K_{Pi}, K_{Ii} and K_{di} are defined as triangular partitions with five segments from 0 to 1 as shown in Fig. 3. Zero (ZO) is the center membership function. The remaining parts of the partition are Negative Big (NB), Negative Small (NS), Positive Small (PS) and Positive Big (PB). The membership function sets for $\Delta\omega_i$, $\Delta(\Delta\omega_i)$ is the same as the Membership Function (MF) sets as shown in Fig. 3.

IV. GRAVITATIONAL SEARCH ALGORITHM

The Gravitational Search Algorithm (GSA) is constructed based on the law of gravity and the notion of mass interactions. GSA is one of the newest heuristic algorithms which have been inspired by the Newtonian laws of gravity and motion. In GSA a set of agents called masses are introduced to find the optimum solution by simulation of Newtonian laws of gravity and motion [13]. Also, each mass agent has four specifications: position, inertia mass, active gravitational mass, and passive gravitational mass. The position of the mass corresponds to a solution of the problem, and its gravitational and inertial masses are determined using a fitness function. In other words, each mass presents a solution, and the algorithm is navigated by properly adjusting the gravitational and inertia masses. By lapse of time, we expect that masses be attracted by the heaviest mass. This mass will present an optimum solution in the search space [14].

The GSA could be considered as an isolated system of masses. It is like a small artificial world of masses obeying the Newtonian laws of gravitation and motion.

More precisely, masses obey the following laws:

- Law of gravity: Each particle attracts every other particle and the gravitational force between two particles is directly proportional to the product of their masses and inversely proportional to the distance between them, R.
- Law of motion: The current velocity of any mass is equal to the sum of the fraction of its previous velocity and the variation in the velocity. Variation in the velocity or acceleration of any mass is equal to the force acted on the system divided by mass of inertia.

To describe the GSA consider a system with s masses in which position of the i_{th} mass is defined as:

$$X_i = (x_i^1, ..., x_i^d, ..., x_i^n), i = 1,2,..., s$$

Where, x_i^d is position of the i_{th} mass in the d_{th} dimension and n is the dimension of the search space. According to [15] mass of each agent is computed after calculating current population's fitness as:

$$M_i(t) = \frac{q_i(t)}{\sum_{j=1}^{s} q_i(t)}$$

Where, $M_i(t)$ is the mass value of the agent i at t.

$$q_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)}$$

Where, $fit_i(t)$ is the fitness value of the agent i at t, and worst (t) and best (t) are defined as follows for the minimization problem:

$$best(t) = \min_{j \in \{1,...,s\}} fit_i(t)$$

$$worst(t) = \max_{j \in \{1,...,s\}} fit_i(t)$$

To compute acceleration of an agent, total forces from a set of heavier masses that apply on it should be considered based on the law of gravity, which is followed by calculation of agent acceleration using the law of motion. Afterwards, next velocity of an agent is calculated as a fraction of its current velocity added to its acceleration (Eq. (8)). Then, its next position can be calculated using Eq. (9):

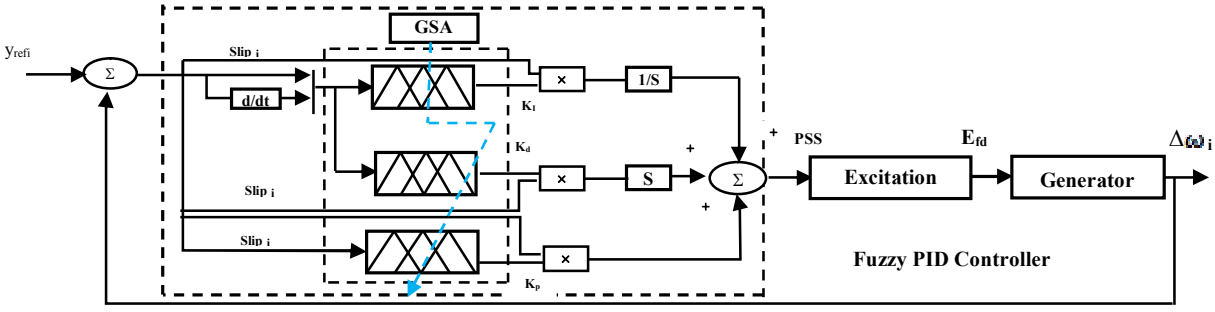


Figure 2. Structure of the proposed GSAF based PSS

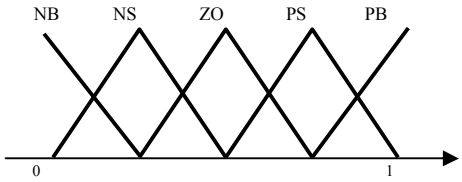


Figure 3. The MF sets for K_{Pi}, K_{Ii} and K_{di}.

$$F_i^d(t) = \sum_{j \in kbest, j \neq i} rand_i G(t) \frac{M_j(t) M_i(t)}{R_{ij}(t) + \varepsilon} (x_j^d(t) - x_i^d(t))$$

$$a_i^d(t) = \frac{F_i^d(t)}{M_i(t)} = \sum_{j \in kbest, j \neq i} rand_j G(t) \frac{M_j(t)}{R_{ij}(t) + \varepsilon} (x_j^d(t) - x_i^d(t))$$

$$V_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t)$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1)$$

Where, $rand_i$ and $rand_j$ are two uniformly distributed random numbers in the interval [0,1], ε is a small value, $R_{ij}(t)$ is the Euclidean distance between two agents i and j, defined as $R_{ij}(t) = ||X_i(t), X_j(t)||_2$, *kbest* is the set of first *K* agents with the best fitness value and biggest mass, which is a function of time, initialized to K_0 at the beginning and decreasing with time. Here K_0 is set to s (total number of agents) and is decreased linearly to1.

In GSA, the gravitational constant, *G*, will take an initial value, G_0, and it will be reduced with time:

$$G(t) = G(G_0, t)$$

Also some differences and advantages of this technique are consisting of [14]:

- In GSA, the agent direction is calculated based on the overall force obtained by all other agents.
- In GSA the force is proportional to fitness value and so agents see the search space around themselves in the influence of force.
- GSA is memory-less and only current position of the agents plays a role in the updating procedure.
- In GSA the force is inversely proportional to the distance between solutions.

Fig. 4 shows the flowchart of the proposed intelligent algorithm. In this paper, evaluation of the Integral of the Time multiplied Absolute value of the Error (ITAE) based function is an alternative of the conventional maximization of fitness function, which defined as follows:

$$f(ITAE) = \frac{1}{1 + MSE(ITAE)}$$

Where,

$$MSE(ITAE) = (\sqrt{\sum_{i=1}^{m} ITAE_i}) / m,$$

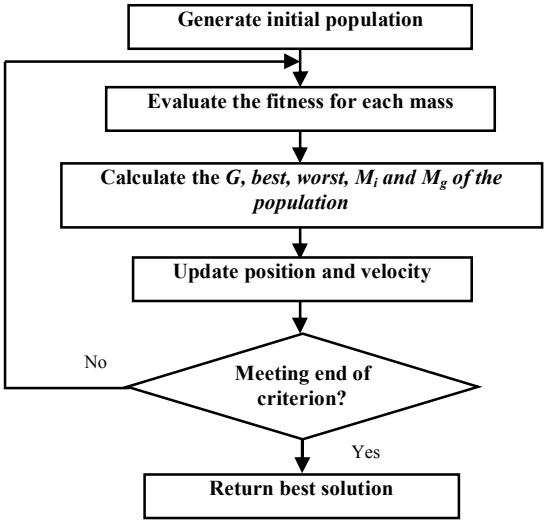


Figure 4. Flowchart of GSA

V. PROBLEM STATEMENT

In this paper, the 10-machine 39-bus New England power system shown in Fig.5 is employed. Furthermore, the system is tested in various operating points as nominal, lightly and heavily conditions. Details of the system data and operating condition are given in [11].

A. GSA-Based Fuzzy PSS Design

The proposed fuzzy PSS is connected to machines G_5, G_7 and G_9 in the test system. Here, the modified GSA evolution procedure is applied to produce the rule Tables of the proposed GSAF PSS to guarantee relative stability and concurrently secure the time domain specifications. The plot of obtained fitness function value is shown in Fig. 6. The Results of fuzzy rule base sets are listed in Tables I-III.

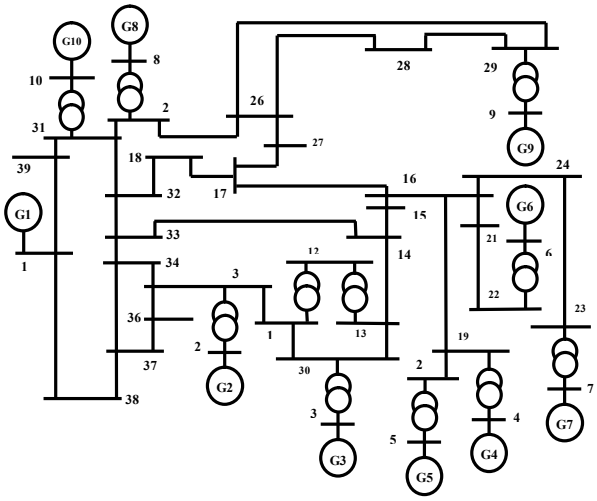


Figure 5. Ten-machine 39-bus New England power system.

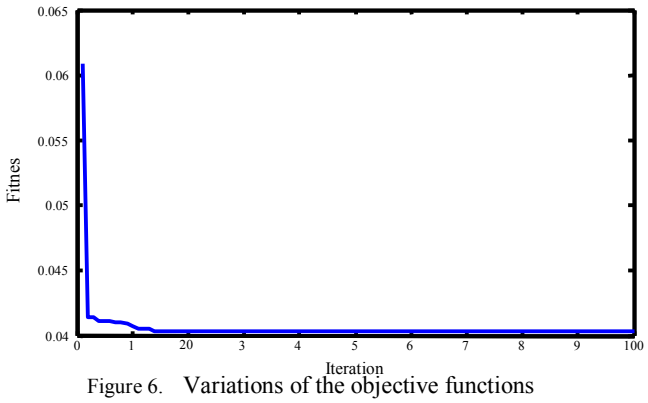


Figure 6. Variations of the objective functions

TABLE I. OPTIMAL RULE BASE FOR K_{PI}

	NB	NS	PS	PB
NB	PM	ZO	PM	PM
NS	NS	ZO	PM	PM
Z	ZO	NM	NM	ZO
PS	ZO	PM	PM	PM
PB	PM	PM	PB	PM

TABLE II. OPTIMAL RULE BASE FOR K_{II}

	NB	NS	PS	PB
NB	ZO	ZO	NM	PB
NS	ZO	PM	ZO	PM
Z	PM	NB	NB	PB
PS	NS	NS	NS	PM
PB	PB	ZO	PM	PM

TABLE III. OPTIMAL RULE BASE FOR K_{DI}

	NB	NS	PS	PB
NB	ZO	ZO	NM	PB
NS	ZO	PM	ZO	PM
Z	PM	NB	NB	PB
PS	NS	NS	NS	PM
PB	PB	ZO	PM	PM

B. *Result of Simulation*

To demonstrate the effectiveness and robustness of the proposed controller, simulation studies are carried out under fault disturbances for two scenarios. The proposed controller is installed for G_5, G_7 and G_9. The performance of the proposed GSAF PSSs is compared to the TS PSS [11] for different operating conditions.

C. *Scenario 1*

It is very important to test the PSS under the loading power factor operating condition. Therefore, a 0.2 p.u. step increase in the mechanical torque was applied at t=1.0. Fig. 7 shows the results of simulation that are tested in heavy load condition. This is obvious that the proposed GSA based fuzzy controller achieves desired level of robust performance under nominal load condition.

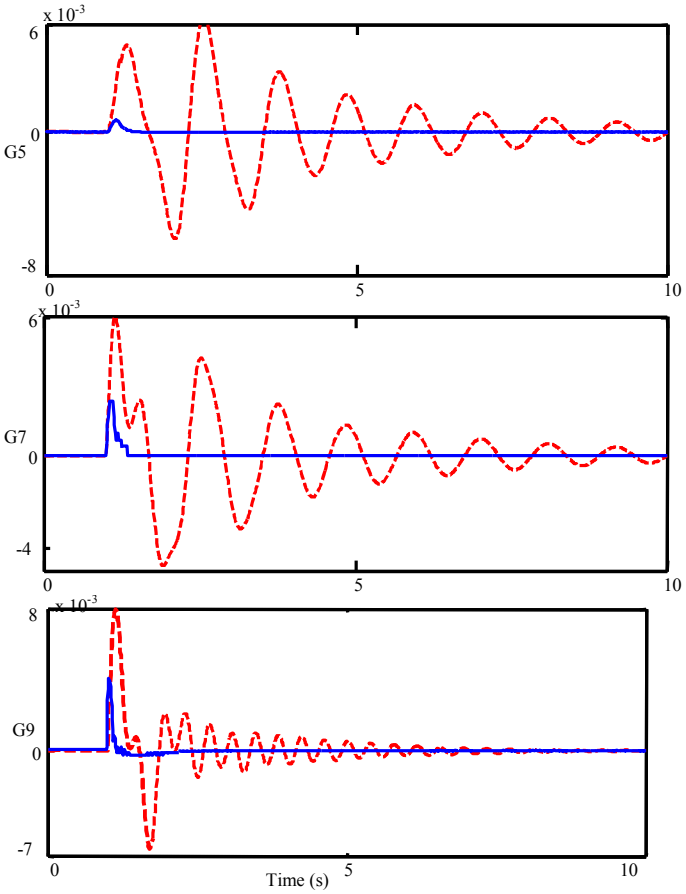


Figure 7. System response under heavy loading in scenario 1, Solid (GSAFPSS) and Dashed (TSPSS).

D. *Scenario 2*

In this scenario a 0.2 p.u. step increase in the mechanical torque was applied at t=1 and after a few seconds a 6-cycle three-phase fault at t=5 sec, on bus 26 at the end of line 26-29 for system will be applied. The results of simulation in nominal load condition are shown in Fig. 8. It can be seen that the overshoot, undershoot, settling time and speed deviations of all machines are greatly reduced by applying the proposed GSA based fuzzy PSSs.

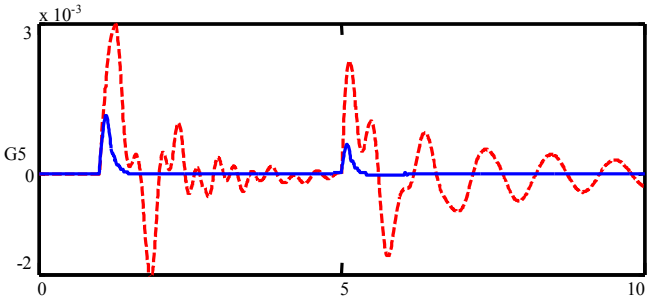


Figure 8. System response under nominal loading in scenario 2, Solid (GSAPSS) and Dashed (TSSS)

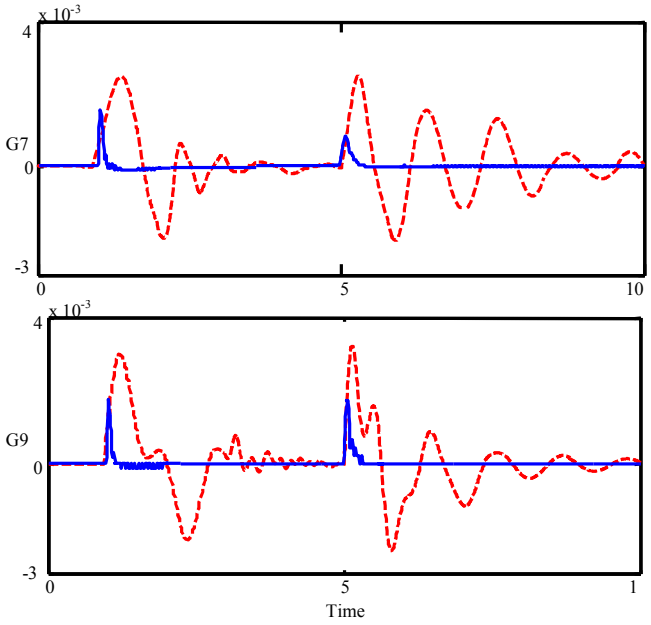


Figure 8. System response under nominal loading in scenario 2, Solid (GSAPSS) and Dashed (TSSS)

VI. CONCLUSIONS

In this paper a new robust GSA based fuzzy controller is proposed for the solution of the low frequency oscillation problem in multi machine power systems in this paper. Fuzzy logic control is one of the most successful areas in the application of fuzzy theory and is excellent alternatives to the conventional control methodology when the processes are too complex for the analysis by conventional mathematical techniques. This newly developed control strategy combines advantage of the fuzzy control system and GSA techniques for achieving the desired level of robust performance under different operating conditions and load disturbances. This new method is stronger than other methods in particular the lack of reliability in what concerns succeeded, and valid convergence, and the failures in attempts to reduce the time. It should be noted that the construction of the optimal rule base sets for the proposed fuzzy PSS is very important to achieve the best performance. Thus, to reduce the fuzzy system effort and cost saving, a modified GSA has been used to produce fuzzy rule Tables. The salient feature of the proposed method is that it does not require an accurate model of the system. The New England 10-machine 39-bus standard power system, under various system configurations and loading conditions, is employed to illustrate the performance of the proposed method. The effectiveness of the proposed controller is compared with TSPSS through some performance indices. Simulation results show the effectiveness of the proposed GSA-based fuzzy, can work effectively over a wide range of the loading conditions and is superior to other compared methods.

REFERENCES

[1] J. Machowski, J. W. Bialek, J. R. Bumby, "Power system dynamics and stability", New York: Wiley; 1997.

[2] P. Kundur, "Power system stability and control", New York: McGraw- Hill; 1994.

[3] N. Tambey, M. Kothari, "Unified power flow controller based damping controllers for damping low frequency oscillations in a power system"; June 2003. IEE Proc. Gen, Transm. Distrib, Vol. 150, No. 2, pp. 129- 140, 2003.

[4] Y. L. Abdel-Magid, M. A. Abido, "Optimal multi-objective design of robust power system stabilizers using genetic algorithms", IEEE Transaction on Power Systems, Vol. 18, No. 9, PP. 256-267, August 2003.

[5] O. Abedinia, M. Salay Naderi, A. Jalili, B. Khamenehpour, "Optimal tuning of multi-machine power system stabilizer parameters using genetic-algorithm", Proceedings of International Conference on Power System Technology, pp. 1-6, , Hangzhou, China, 24-28 Octobr, 2010.

[6] L. Davis, "Genetic algorithms and simulated annealing", Pitman, 1987.

[7] J. H. Hwang, D. W. Kim, J. H. Lee and Y. J. An, "Design of fuzzy power system using adaptive evolutionary algorithm", Engineering Applications of Artificial Intelligence, Vol. 21, pp. 86-96, 2008.

[8] M. Caner, N. Umurkan, S. Tokat, S. V. Ustun. "Determination of optimal hierarchical fuzzy controller parameters according to loading condition with ANN", Expert systems with applications. Vol. 34, pp. 2650-2655, 2008.

[9] L. L. Houn, S. H. Ye, C. Jian, G. Xib-pin, "Novel fuzzy predictive pid control for a class of nonlinear systems", Proceeding of 2002 International Conference Machine Learning and Cybernetics, Vol. 1, No. 2, PP. 293-297, 2002.

[10] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi. "GSA: A gravitational search algorithm". Information Sciences. Vol. 179, No. 13, pp. 2232-2248, 2009.

[11] YL. Abdel Magid, MA. Abido, AH. Hantawy, "Robust tuning of power system stabilizer multi-machine power systems", IEEE Trans. Power Syst. Vol. 15, No. 2, pp. 735-740, 2000.

[12] H. A. Shayanfar, H. Shayeghi, O. Abedinia, A. Jalili. "Design rule-base of fuzzy controller in multi-machine power system stabilizer using genetic algorithm". In: Proceedings of the international conference on artificial intelligence, July 12-14, Las Vegas, Nevada, USA, pp. 43-49, 2010.

[13] A. Chatterjee, G. K. Mahanti. "Comparative performance of gravitational search algorithm and modified particle swarm optimization algorithm for synthe- sis of thinned scanned concentric ring ar- ray antenna". Progress In Electromagnetics Research B, Vol. 25, pp. 331-348, 2010.

[14] E. Rashedi , H. Nezamabadi-pour, S. Saryazdi, MM. Farsangi, "Allocation of static var compensator using gravitational search algorithm", First Joint Congress on Fuzzy and Intelligent Systems, Ferdowsi University of Mashhad, 29-31 Aug 2007.

[15] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, "Filter modeling using gravitational search algorithm". Engineering Applications of Artificial Intelligence. Vol. 24, No. 1, pp. 117-122, 2010.

BIOGRAPHIES

Heidarali Shayanfar received the B.S. and M.S.E. degrees in Electrical Engineering in 1973 and 1979, respectively. He received his Ph. D. degree in Electrical Engineering from Michigan State University, U.S.A., in 1981. Currently, he is a Full Professor in Electrical Engineering Department of Iran University of Science and Technology, Tehran, Iran. His research interests are in the Application of Artificial Intelligence to Power System Control Design, Dynamic Load Modeling, Power System Observability Studies, Voltage Collapse, Congestion Management in a Restructured Power System, Reliability Improvement in Distribution Systems and Reactive Pricing in Deregulated Power Systems. He has published more than 350 technical papers in the International Journals and Conferences proceedings. He is a member of Iranian Association of Electrical and Electronic Engineers and IEEE.

Oveis Abedinia received the B.S. and M.Sc. degrees in Electrical Engineering from Azad University, Ardabil and Science and Technology Research Branch, Tehran, Iran in 2005 and 2009, respectively. His areas of interest in research are Application of Robust Control, Artificial Intelligence to Power System Control and restructuring, optimization in Distribution Generation and power system.

Mohammad Salay Naderi received his B.Sc. in electric power engineering (1996) and M.Sc. (1998) in high voltage engineering from Sharif University of Technology in Tehran, Iran. In 2006, he finished his PhD career at the same university which was on partial discharge monitoring and localization within power transformer. He joined Energy Systems Research Group in the School of Electrical Engineering and Telecommunications at UNSW in 2004 as Visiting Research Associate and continued his research until late 2005. He worked with Iran Power Generation and Transmission Company (Tavanir Holding Co.) as Deputy General Manager of Transmission System Technical Administration Bureau from 2005 to 2010. In 2010 he joined the School of Electrical Engineering and Telecommunications at UNSW as Lecturer. His research and teaching interests are in high voltage engineering and condition monitoring of power equipment; sustainable energy systems analysis and development as well as distributed generation and its effects of power system protection.

Ali Ghasemi received the B.S. degree in Electrical Engineering from Esfahan University of Technology, Esfahan, Iran in 2009. Currently, he is a M.S.E. student in Technical Eng. Department of the University of Mohaghegh Ardabili, Ardabil, Iran. His Areas of interest in Research are the Application of Heuristic Optimization to Power System Control.

Applying Fuzzy Image Processing Technology to Inspect Defects of Thin Film Transistor-Liquid Crystal Display

Chung-Feng Jeffrey Kuo[1], Chin-Hsun Chiu[2], Tai-Yuan Su[1], Kai-Ching Peng[1]

[1]Graduate Institute of Automation and Control, National Taiwan University of Science and Technology, Taipei, Taiwan, Republic of China
[2] Department of Materials Science and Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan, Republic of China
E-MAIL: jeffreykuo@mail.ntust.edu.tw, D9404302@mail.ntust.edu.tw, M9612001@mail.ntust.edu.tw, D9912001@mail.ntust.edu.tw

Abstract- *TFT-LCD cell process the grinding process, which can smoothen out the even edges of the TFT-glass substrate. In this processing, edges of the TFT-glass substrate easily fracture, which may damage the circuit of terminals. By using AOI to inspect the terminal, this study designs a fuzzy operator to separate the defect based on the arc curve that the detect features. By introducing the concept of image pyramid into the fuzzy operator, this study enables this fuzzy operator to be suitable for detecting defects, which are extremely small in size. Experimental results proved that the AOI for this study can accurately decide the grinding position of the TFT-glass substrate and simultaneously inspect terminals to ensure the terminal integrity.*

Keywords: TFT-LCD, AOI, Fuzzy operator, Image pyramid

1. Introduction

In the Thin film transistor-liquid crystal display (TFT-LCD) cell processing, a TFT-glass substrate needs to be cut for determining its size and the grinds the edges after the TFT-glass substrate is cut. The TFT-glass substrate is composed of glass layer, color filter, and film transistors; circuits on terminal are included on the edges of the TFT-glass substrate. If positional deviation, fracture, and pollutants arise between respective laminates, terminals on the panel will be rendered short and thus malfunction. Moreover, the edges may be fractured or damaged in grinding process. This causes defective finished TFT-LCD products, which need to be repaired or directly destroyed. This paper focuses on the inspection of the edges of the TFT-glass substrate to ensure TFT-LCD production quality.

This paper is organized as follows: In Section 2, a survey on edge detection method for detection of defects is provided. Section 3 presents the proposed method. Experimental results are discussed in Section 4. Finally, the concluding remarks are given in the last section.

2. Related work

For the past few years, the Automatic Optical Inspection (AOI) has been widely used for defect detection in product processing. Liu et al. [1] presented an inline defect-detection system for a TFT-LCD array process. Tsai [2,3] proposed the correlation coefficient for identification of defects on a printed circuit board (PCB).

For edge detection of image process, the method proposed by Sobel and Feldman [4] and Canny [5] are used generally. In terms of target edge detection, Lewis [6] proposed the rapid correlation coefficient method. In this method, there was no need for normalization, and operation was accelerated. Tao [7] proposed the application of fuzzy rules to determine the edges of an object, each pixel in the mask is represented as SMALL and LARGE by membership function. In this method, the threshold does not need to be set in advance. Russo [8] applied the genetic algorithm to fuzzy based edge detection to broaden the range of edge detection application. Russo and Ramponi [9] reduced the fuzzy rules to reduce the detecting time. Russo and Russo and Ramponi [10] proposed a new edge detection method in which the brightness difference is added in the membership function and interference that noises on the fuzzy operator are decreased. Lee and Chain [11] used the image pyramid in the rapid vector quantization method for improve efficiency on image detection.

As the defect of the TFT-glass substrate in this study has the feature of arc curve, Russo's fuzzy operator characteristics are used and this fuzzy operator is improved to be able to detect the arc edges. The distance information is added to the fuzzy algorithm, and combined with the characteristics of the image pyramid to enable the fuzzy operator to detect defects within a relatively wide range. By utilizing the image of the inspected object which is taken by the charge-coupled device (CCD), and marking and separating the defect detected from the original image with the designed algorithms, this study provides basis for inspectors to evaluate the overall quality. Sample images for this study are collected from automatic optical platform.

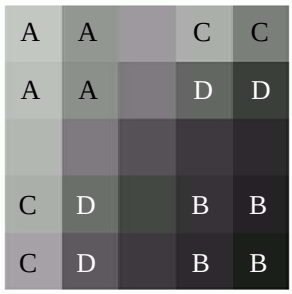
This optical platform is specially designed for TFT-LCD defect detection, and provides good accuracy, resolution and light source quality for edge defect detection. It has good control over the vibration in image capturing.

3. Research methods

This section presents in detail the arc curve fuzzy operator and the Image pyramid for detection of edge defect.

3.1 Arc curve fuzzy operator

The arc curve fuzzy (ACF) operator designed by this study includes a 5 × 5 mask, and arc curve analysis and fuzzy algorithm system. The data obtained from the fuzzy algorithm is used to decide whether the region includes a defect. By observing the characteristics of the arc curve of which the defect image consists which are shown by Figure 1(a), we find that two corners A and B symmetrical with each other feature obvious difference between light and dark and the brightness in region C is greater than that in region D. An arc curve features these characteristics, so the existence of these characteristics can prove that a curve exists on the image.



(a)

P1	P6	P11	P16	P21
P2	P7	P12	P17	P22
P3	P8	P13	P18	P23
P4	P9	P14	P19	P24
P5	P10	P15	P20	P25

(b)

Figure 1 Arc curve fuzzy operator

Next, we further discuss how to discriminate light from dark. A pixel value represents the absolute brightness and a large value does not out of question indicate the light surface because the relative brightness obtained from the comparison is needed for deciding a light surface, that is, there is a probability that an arc curve exists when the brightness in region A is low and the brightness in region B is even lower. In light of this case, the light and shade is represented as fuzzy set in this study.

Numerical values in regions A and B on Figure 1(a) are taken out and the difference between the means of numerical values from regions A and B. This difference is the information of the relative brightness. Such information enables the fuzzy system to preliminarily decide the distribution of the edges. However, for describing the features of the edges, another condition is needed, that is, the brightness in region C must be greater than that in region D and the difference between region C and region D. Under the circumstance that these conditions are met, there exists quite a big possibility that an arc curve exists under this 5 × 5 mask.

The discrimination of the relative brightness value must depend upon the distance relationship between pixels, that is, the definition for relative brightness between region A and region B is different from that for relative brightness between region C and region D. As the distance between A and B is large, it can be deemed as a large relative brightness only when brightness difference is large. However, the distance difference between block C and block D lies in only one or two pixels. In case of such small distance, a relative brightness which does not need to be very high should be deemed as the membership of the light range. To distinguish this issue, this study introduced distance information in the fuzzy system and such distance information can discriminate the light from dark more specifically. Therefore, this study classified image data from the 5 × 5 mask into group A, B, C, D and E based on fuzzy operator and gave explanation to them respectively, as shown by Figure 1(b). The groups follow by

$$A = (p1+p2+p6+p7)/4-(p19+p20+p24+p25)/4 \quad (1)$$

$$B = (p4+p5)/2-(p9+p10)/2 \quad (2)$$

$$C = (p16+p21)/2-p(17+22)/2 \quad (3)$$

$$D=|p4-p5| \quad (4)$$

$$E=|p16-p21| \quad (5)$$

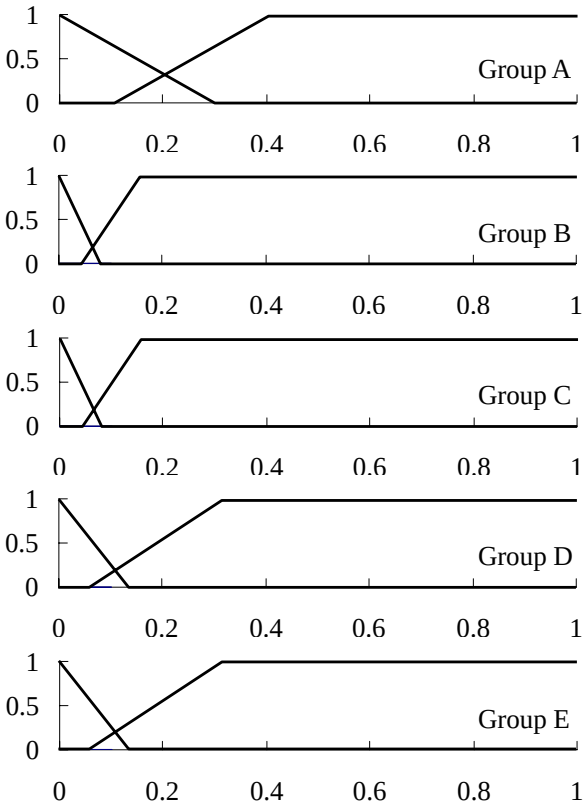


Figure 2 Membership function of each group

Therefore, the membership function for each group are shown by Figure 2, the fuzzy rule are shown as follow:

1. If A is Large and B is Large and C is Large and D is small and E is small then defect.
2. If A is not Large and B is not Large and C is not Large and D is not small and E is not small then clear.

Finally, center of gravity method is used in this study for defuzzification.

3.2 Image pyramid

The principle of image pyramid is that each 4 pixels are taken as a group from the original image at the bottommost layer and each group is averaged and the resulted mean is taken as the next layer, as shown in Figure 3. The higher the layer is, the fewer the pixels are and the lower the resolution is, however the image size can be effectively reduced. By introducing the concept of mean, the appropriate scale invariant feature can be retained between respective layers.

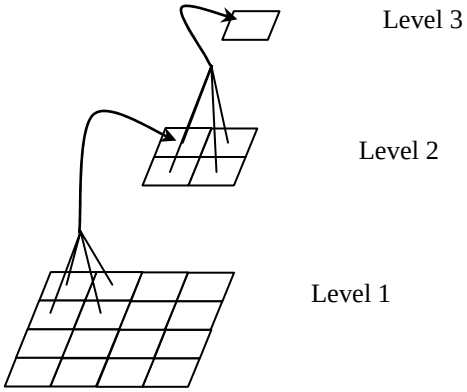


Figure 3 Architecture of the Image pyramid

As the ACF operator is an algorithm which mainly works to discriminate the shape, the shape covered by the mask will affect the discrimination of the defect, which is shown by Figure 4. On this figure, the circle at left indicates the size of the assumed defect and such defect edge is too large for the mask and the curve feature is not obvious. The graphic at right is a graphic scaled down by the image pyramid and the defect scaled down can indicate the curve features better.

Although the defect size affects the decision provided by the fuzzy operator, there still exists a very great probability to discover the defect because the defect includes various edges with different curvatures. Also as the ACF operator has relatively high computation complexity, the mask is not too large. So, such restriction causes the defect to be too large or too small relative to the 5×5 mask and a problem of failure in detection will arise. However, by taking advantage of the interested defect range and applying the image pyramid for search, over too small defect can be avoided from being taken into calculation. Therefore, the selection range for defect may change as appropriate and features better adjustability for detection, so the image pyramid is used to scale down the defect to be detected which is smaller than the mask into the corresponding ACF operator. This can result in the success in detection of the defect within such range and also reduce the interference caused by noises and amount of calculation.

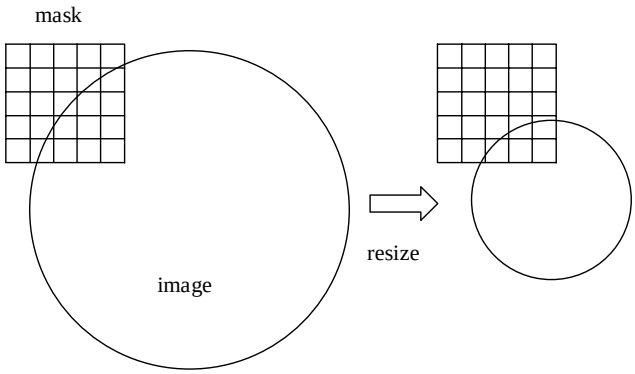


Figure 4 Arc curve fuzzy operator on scaled image

4. Experimenal results

The ACF operator mentioned previously is used to respectively verify the functions, which include verification of ACF operator, verification of smoothing mask effect, scale variation's effects on the ACF operator, etc.

4.1 Variation for detection of edge defect

In the experiment, samples having or not having defects are employed to verify detection effects and finally represent the experimental results with the sum. The experiment is designed and experimental samples are used to test the ACF operator. Various samples having edge defects are tested and include terminals, connected wires between terminals and blocks on LCD panel edges, wherein, the regions have defects or do not have defects and the results will be taken into statistics and represented with false acceptance rate (FAR) and false rejection rate (FRR). Samples are classified into two categories and 50 samples have defects and another 50 samples do not have defects. The regions selected from the samples include those with complicated lines, line intersections, large and small brightness variation regions, as shown in Figure 5. The samples with defects should be selected on a single-sample and single-defect basis to determine the detection target. All sample images have a resolution of 80×80 pixels.

In the experiment, all samples with defects are mixed with those without defects and the fuzzy operator is used for detection. The fuzzy operator is employed to separate images with defects from those without defects and finally the obtained results are analyzed: 4 samples with defects are not detected, so the FAR is 8%. 4 of 50 images without

defects are wrongly decided as those with defects, so the FRR is 8%. Totally 8 of 100 samples are wrongly identified, so the total recognition rate is 92%. Figure 6 shows the experimental result of the fuzzy operator is used for detection.

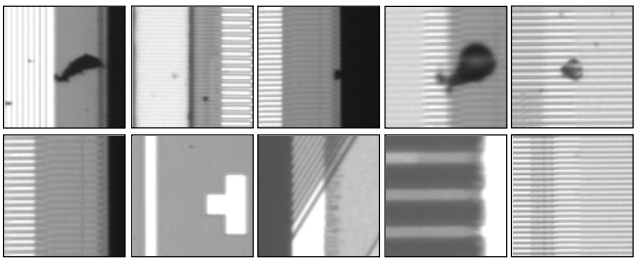


Figure 5 The defect images of edge defect

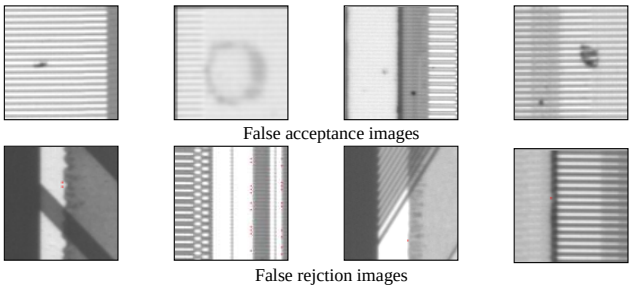


Figure 6 Defect detect results by arc curve fuzzy operator

4.2Verification of effects of smoothing mask

All images with defects are mixed with those without defects and the ACF operator is used for detection. Experimental results are shown on Figure 7. On this figure, (A) is the image which is wrongly decided by the ACF operator, wherein, (a), (b) and (c) are false rejection images and (d) and (3) are false acceptance images. As seen, (B) is the results after the smoothing mask is added. Originally falsely rejected (a), (b) and (c) images all pass the test, which are as shown by (f), (g), (h). Originally falsely rejected (d) and (e) also pass the test, which are shown by (i) and (j).

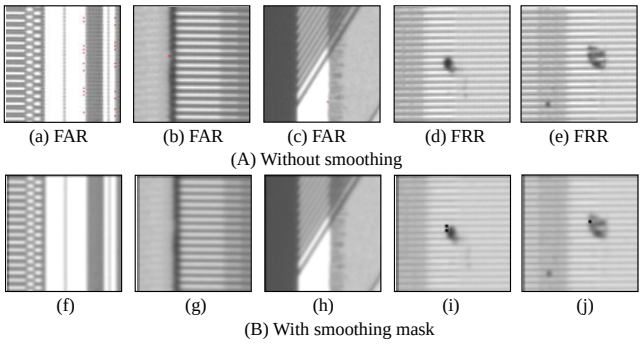


Figure 7 White defect test results

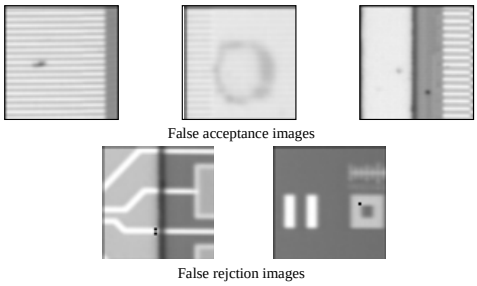


Figure 8 White defect recognition results

Of 50 images with defects and 50 images without defects, 3 with defects are not detected, so the FAR is 3%. This FAR is better than 4% produced in the case that no smoothing mask is used. Among samples, 2 without defects are wrongly decided as those with defects, so the FRR is 2%, and such FRR is better than 4% in the case that no smoothing mask is used. As 5 of 100 sample images can not be correctly recognized, the total recognition rate is 95%. From this experiment, it can be found that the smoothing mask is greatly helpful to the increase of the recognition rate of the ACF operator. Figure 8 and Table 1 shows the experimental result.

Table 1 Comparison of experimental result

Methods	FAR	FRR	Recognition rate
Without smoothing	4	4	92%
With smoothing mask	3	2	95%

4.3 Scale variation's effects on fuzzy operator

The purpose of this experiment is to test and verify the defect detection range of the ACF operator. After the perfect circle test image is scaled by the image pyramid, the ACF operator is applied and the results are verified and tested. As the edge curvature of the perfect circle only correlates to the radius, the equivalent scaling will change the curvature of the circle. Therefore, the detection limit can be decided by the circle radius of the minimum detectable edge. By reversely deducting such process, the image pyramid transformation can be applied to detect the defect range or the scale can be changed based on the variation of resolution. In this experiment, circles of various sizes are designed for simulating the effects of the image pyramid and comparing the effects that the image pyramid scale has on the ACF operator with those on the ACF operator after the image pyramid scale transformation. Black circles with a diameter of 300 pixels are taken as samples and scaled down for 30 times, scaled down by 1/2 for the first time, scaled down to 1/3 for the second time and the rest may be deducted by analogy until they are scaled down to 1/30. In the experiment, the ACF operator is applied to 30 samples in order to search for the curves and the arcs with curves searched will be marked, and the circle with marked arc represents the applicable curvature

range of the ACF operator.

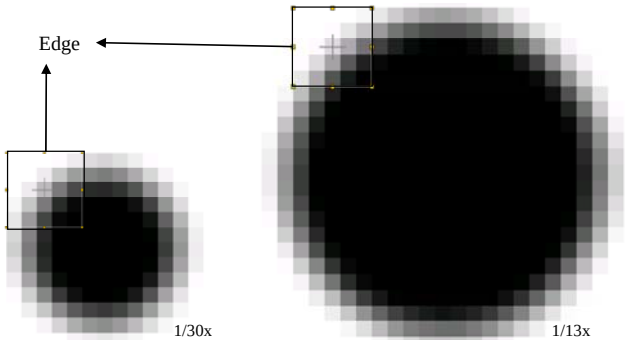


Figure 9 Test samples

In Figure 9, the ratio ranging from 1/13 to 1/30 fails to help the edge detection. The cause lies in the following two points: first, the edge represented by 1/13 tends to be straightened due to scaling and the originally arc edge is represented by the straighter black line and its periphery is filled up by gray blocks with different gray scales to result the image to take a circular shape between the digital gap. Therefore, the defect can be detected as long as it has a relatively large curvature. Second, a too small circle fails to fill up the 5×5 mask of the ACF operator, so it can not be used to recognize the arc. However, circles including 300 to 25 pixels all can be recognized by the ACF operator and still most of the circles having defect size and arc curvature lie within the range from 300 to 25 pixels. The image pyramid can be applied to the defects having less than 25 pixels and the ACF operator can be used for detection by scaling up the image. If the image file to be detected is too large, the image can be scaled down with the image pyramid. As long as the minimum allowable defect range is known, the image file can be reduced to the appropriate size to increase the calculation efficiency.

4.4 Discussion

In the experiment for verification of ACF operator, the ACF operator turned out excellent results in which the FAR is 4% and the FRR is 4%. In the experiment for verification of smoothing mask effects, the FAR decreased to 3% and the FRR decreased to 2%. Finally, this paper discussed the scale variation's effects on the ACF operator to prove that the applicable range of the ACF operator is extremely wide and the image pyramid can be jointly used to extend the scale on demands to detect defects which are extremely small.

5. Conclusion

This study proposed theories and methods for inspecting TFT-glass substrate, and the experiments proved that the theories are able to detect defects successfully. And this study provided various defects distributed on terminals on the TFT all can be detected by the arc curve fuzzy operator in the edge defect detection method, which indicates that the arc curve fuzzy operator has excellent defect detection capability and has an extremely wide applicable range. Finally, by taking experimental results into statistics, we found that the total recognition rate of the arc curve fuzzy operator reached 95%. Relying on the grinding detection positioning method and edge defect detection method proposed by this study, the complete AOI for TFT cell grinding process can be provided to improve the manufacturing quality of TFT.

Acknowledgement

The research was supported by the National Science Council of the Republic of China under the grant No. 98-2221-E-011-013-MY3.

References

[1] Liu Y H, Huang Y K and Lee M J 2008 Automatic inline defect detection for a thin film transistor–liquid crystal display array process using locally linear embedding and support vector data description, *Meas. Sci. Technol.*19 095501

[2] Tsai D M and Lin C T 2003 Fast normalized cross correlation for defect detection, *Pattern Recognit. Lett.* 24 2625-31

[3] Tsai D M and Lin C T and Chen J F 2003 The evaluation of normalized cross correlations for defect detection, *Pattern Recognit. Lett.s* 24 2525-35

[4] Duda R O. and Hart P E 1973 *Pattern classification and scene analysis* (New York: Wiley-Interscience Publication)

[5] Canny J 1986 A computational approach to edge detection, *IEEE Trans. Pattern Anal. Mach. Intell.* 8 679-98

[6] Lewis J P 1995 *Fast template matching* (Canada: Canadian Image Processing and Pattern Recognition Society)

[7] Tao C W, Thompson W E, and Taur J S 1993 A fuzzy if then approach to edge detection, *IEEE Internationl Conference on Fuzzy Systems, FUZZ-IEEE 93* 1356-60

[8] Russo F 1993 A new class of fuzzy operators for imageprocessing: design and implementation, *IEEE Internationl Conference on Fuzzy Systems, FUZZ-IEEE 93* 815-820

[9] Russo F and Ramponi G 1994 Nonlinear fuzzy operators for image processing, *IEEE Trans. Signal Processing* 38 429-40

[10] Russo F and Ramponi G 1994 Edge extraction by FIRE operators, *IEEE Iinternationl Conference on Fuzzy Systems, FUZZ-IEEE 94'* 249-53

[11] Lee C H and Chen L H 1995 A fast search algorithm for vector quantization using mean pyramids of code words, *IEEE Trans. Commun.* 43 1697-702

Genetic-PSO Fuzzy Data Mining With Divide and Conquer Strategy

Amin Jourabloo
Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
E-mail: jourabloo@ce.sharif.edu

Abstract - *Nowadays, discovery the association rules is an important and controversial area in data mining research studies. These rules, describe noticeable association relationships among different attributes. While most studies have focused on binary valued transaction data, in real world applications, there data usually consist of quantitative values. With that in mind, in this paper, we propose a fuzzy data mining algorithm for extracting membership functions from quantitative transactions. This is a hybrid genetic-pso algorithm for finding membership functions suitable for mining problems by a strong cooperation of GA and PSO. This algorithm integrates the two techniques entire run of simulation in each iteration, a part of population are substituted by new ones generated by means of GA, while the other part is the same of previous generation but moved on the solution space by PSO. At the end, best final sets of membership functions in all the populations are gathered to be used for mining fuzzy association rules. According to experimental results, the proposed genetic-pso fuzzy data mining algorithm has a good effect on fitness of membership functions.*

Keywords: data mining, fuzzy sets, genetic algorithms (GA), Particles Swarm Optimization (PSO), membership functions.

1 Introduction

An important area of data mining research deals with the discovery of association rules, which describe interesting association relationships among different attributes. [5] Association rule techniques are generally applied to databases of transactions where each transaction consists of a set of items. [6] Let us consider a database of customer transactions *T,* where each transaction is a set of items. The objective is to find all rules of the form $X \Rightarrow Y$, which correlate the presence of one set of items *X* with another set of items *Y.* An example of such a rule is:

98% of people who purchase diapers and baby food also buy baby soap. [5]

Most previous studies have focused on binary valued transaction data. Transaction data in real-world applications, however, usually consist of quantitative values. Designing a sophisticated data-mining algorithm able to deal with various types of data presents a challenge to workers in this research field. Recently, fuzzy set theory has been used more and more frequently in intelligent systems because of its simplicity and similarity to human reasoning. The theory has been applied in fields such as manufacturing, engineering, diagnosis, economics, among others.

Several fuzzy learning algorithms for inducing rules from given sets of data have been designed and used to good effect with specific domains [7]. Evolutionary computing (EC) is an exciting development in mining algorithms. It amounts to building, applying and studying algorithms based on Darwinian principles of natural selection [1, 2]. Genetic algorithms (GAs) are a family of computational models developed by Holland [3, 4].Genetic Algorithms (GA) and Particles Swarm Optimization (PSO) are both population based algorithms that have proven to be successful in solving a variety of difficult problems. However, both models have strengths and weaknesses. Comparisons between GAs and PSOs have been performed by Eberhart and Angeline and both conclude that a hybrid of the standard GA and PSO models could lead to further advances. [9,10] Several hybrids Genetic, Particle Swarm Optimization algorithms have been designed .As to [8] proposed an algorithm that combines the standard velocity and position update rules of PSOs with the ideas of selection, crossover and mutation from GAs. The algorithm is designed so that the GA facilitates a global search and the PSO performs a local search. [4]

In 2006 Esmin, et al [11] proposed a new model called Hybrid Particle Swarm Optimizer with Mutation (HPSOM), by integrate the mutation process often used in GA into PSO. This process allows the search to escape from local optima and search in different zones of the search space.

This paper, thus, proposes a fuzzy data-mining algorithm for extracting both association rules and membership functions from quantitative transactions. A hybrid genetic-pso algorithm for finding membership functions suitable for mining problems is proposed that consists in a strong cooperation of GA and PSO, since it maintains the integration of the two techniques for the entire run of simulation. In each iteration, in fact, some of the individuals are substituted by new generated ones by means of GA, while the remaining part is the same of the previous generation but moved on the solution space by PSO. Considering Genetic Algorithms and Particle Swarm Optimization algorithms, most of the times, PSO has faster convergence rate than GA initially, but they are often outperformed by GA for long simulation runs.

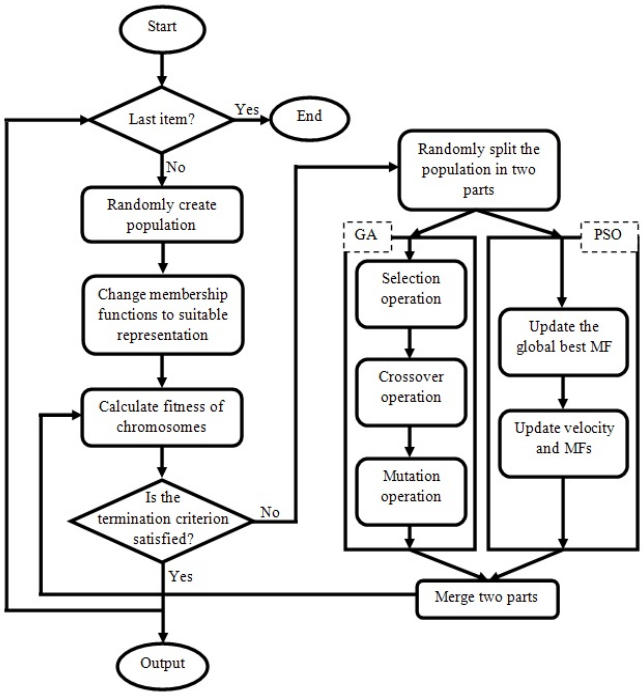


Fig. 1. The proposed GA-PSO flowchart for fuzzy data mining.

2 GA-PSO Mining framework based on the divide-and-conquer strategy

In this section, the fuzzy and GA-PSO concepts are used to discover both useful association rules and suitable membership functions from quantitative values. A GA-PSO framework with the divide-and-conquer strategy is proposed for searching for membership functions suitable for the mining problems. The final best sets of membership functions in all the populations are then gathered together to be used for mining fuzzy association rules. The proposed framework is shown in Fig. 1. The proposed framework in Fig. 1 is divided into two phases: mining membership functions and mining fuzzy association rules. Assume the number of items is. In the phase of mining membership functions, it maintains populations of membership functions, with each population for an item. Each chromosome in a population represents a possible set of membership functions for that item. The chromosomes in the same population are of the same length. The proposed mechanism then chooses appropriate strings for "mating," gradually creating good offspring sets of membership functions. The offspring sets of membership functions undergo recursive "evolution" until a good set of membership functions has been obtained. Next, in the phase of mining fuzzy association rules, the sets of membership function for all the items are gathered together and used to mine the fuzzy interesting association rules from the given quantitative database.

2.1 Genetic Algorithm

Genetic algorithms were first introduced by Holand in the early 1970's [3] and have been widely successful in optimization problems. The genetic operator that used in this paper have been described in [7] and [17].In representation step, each set of membership functions are encoded as a chromosome and handled as an individual with real-number schema. Genetic operators are very important to the success of specific GA application. The crossover and mutation operators chosen in this paper are the max-min-arithmetical (MMA) crossover proposed in [17] and the one-point mutation proposed in [7].

2.1.1 Fitness

The fitness value of each set of membership function is determined according to two factors: suitability of membership functions and fuzzy supports of large 1-itemsets that have been described in [7].
The fitness value of a chromosome C_q is defined a

$$f(C_q) = \frac{\sum_{X \epsilon L_1} fuzzy\_support(X)}{suitability(C_q)} \quad (1)$$

where L_1 is the set of large 1-itemsets obtained by using the set of membership function in C_q

2.2 PSO

The Particle Swarm Optimization (PSO) algorithm is a new optimization algorithm inspired by social behavior in nature. Like Genetic Algorithms, the PSO is a population-based optimization method that searches multiple solutions in parallel. However PSO employs a "cooperative" strategy unlike GA, which utilizes a "competitive" strategy.

During each generation each particle is accelerated toward the particle's previous best position and the global best position. At each iteration a new velocity value for each particle is calculated based on its current velocity, the distance from its previous best position, and the distance from the global best position. The new velocity value is then used to calculate the next position of the particle in the search space. This process is then iterated a set number of times or until a minimum error is achieved.
In the inertia version of the algorithm an inertia weight, reduced linearly each generation, is multiplied by the current velocity and the other two components are weighted randomly to produce a new velocity value for this particle, this in turn affects the next position of the particle during the next generation. Thus, the governing equations are:

$$V_i(t) = \omega\, V_i(t\text{-}1) + c_1\, \phi_1\, (P_i - X_i(t\text{-}1)) + c_2\, \phi_2\, (P_g - X_i(t\text{-}1))$$

$$X_i(t) = X_i(t\text{-}1) + V_i(t) \quad (2)$$

Where in this paper, t is number of generation and X_i is particle i's membership function, V_i is particle i's velocity, P_i is particle i's previous best membership function and P_g is the global best particles membership function. The parameter ω is the inertia weight and variables c_1, c_2, ϕ_1 and ϕ_2 are social parameters and random numbers in the range [0.0, 1.0], respectively.

2.3 GA – PSO

GA and PSO are much similar in their inherent parallel characteristics, whereas experiments show that they have their specific advantages when solving different problems. What we would like to do is to use both their excellent features by synthesizing the two algorithms

3 The proposed mining algorithm

The input is a body of *n* quantitative transaction data, a set of *m* items, each with a number of predefined linguistic terms, a support threshold α, and a population size *P* and we are looking for the output a set of membership functions for extracting association rules. First randomly generate *m* populations, each for an item, each individual in a population represents a possible set of membership functions for that items and encode each set of membership functions into a string representation. For each chromosome in each population calculate the fitness value and randomly divide the population into two parts. Execute GA Algorithm on part one using the selection operation to choose membership functions. Any selection operation, such as the elitism selection strategy or the roulette selection strategy may be used here. Now execute crossover operations and then mutation operations on each population. For part two execute PSO Algorithm by finding the best membership function in each population and updating the global best membership function and then update velocity and membership functions. Now merge two parts and create new population, calculate the fitness value of each chromosome in each population. If the termination criterion is not satisfied, again divide this new population into two parts and execute GA and PSO as mentioned on them otherwise gather the sets of membership functions, each of which has the highest fitness value in its population. Note that the termination criterion may be number of iterations, allowed execution time or convergence of the fitness values. The proposed mining algorithm illustrated in Fig. 1

3.1 An Example

In this section, an example is given to show advantage of the proposed mining algorithm. This is a simple example to show how the proposed algorithm can be used to mine membership from quantitative data. Assume that we have one item in a transaction database: Milk and the data set include the six transactions. Assume item has three fuzzy regions: *Low*, *Middle*, and *High*. Thus, three fuzzy membership functions must be derived for item. The population is randomly generated. Assume the population size is ten in this example.

For comparison the proposed algorithm with GA and PSO the best fitness of each generation with GA, PSO and GA-PSO are shown in Fig2

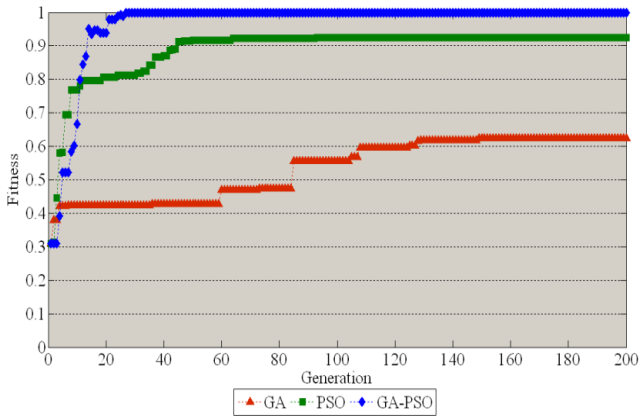


Fig. 2. Comparison the proposed algorithm with GA and PSO

For statistical analysis of proposed mining algorithm the result of 32 runs of GA-PSO and GA are shown separately in Fig3, Fig4 respectively. (Min Max Average)

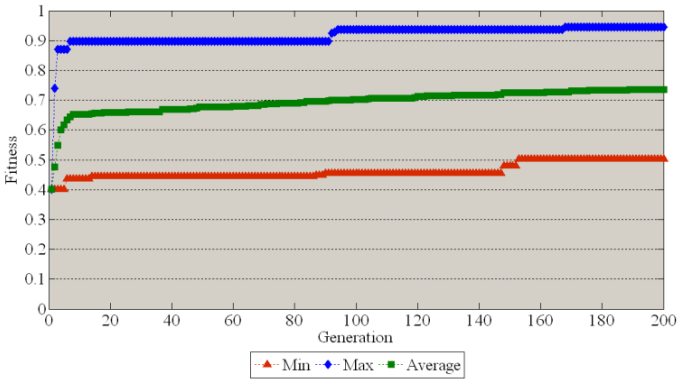


Fig. 3. Statistical analysis for 32 runs of GA

4 Experimental Results

In this section, experiments conducted to show the performance of the proposed approach are described. They were implemented in Matlab on an Intel Core 2 personal computer with 2.00 GHz and 1 GB RAM. The initial population size *P* was set at 10, the crossover rate P_c was set at 0.8, and the mutation rate P_m was set at 0.01 according to [18]. The parameter *d* of the crossover operator was set at 0.35 according to [17], the parameter of the mutation operator was set at 3, the minimum support α was set at 0.04 (4%), the inertia weight ω was set at 0.86 and the social parameters c_1, c_2 are set 0.2.

Simulated datasets with 3 items and with different dataset sizes from 5 to 20 k transactions were used in the experiments. In Table.1, the numerical result of execution of GA, PSO and GA-PSO on different size of database from 5 to 20 k are shown and the result shows that GA-PSO have the best performance.

The relationship between the Average Fitness and the database size is shown in Fig. 5 and the relationship between the execution time and the database size is shown in Fig. 6

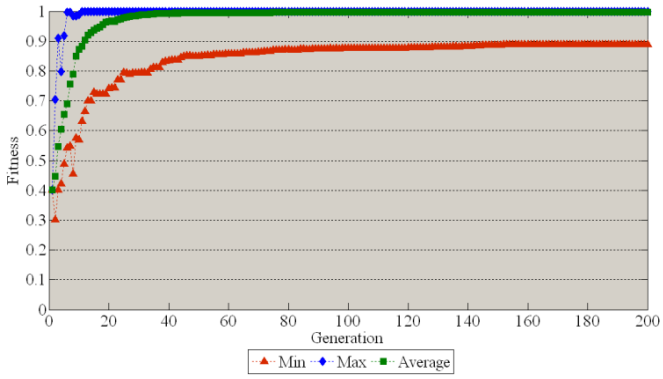


Fig. 4. Statistical analysis for 32 runs of GA-PSO

Table 1

Algorithm (size)	Generation	Fitness1	Fitness2	Fitness3	Average Fitness	Time(minute)
GA(5 KB)	100	0.40	0.75	0.49	0.54	5
PSO(5 KB)	100	0.74	1	0.70	0.81	3
GA-PSO(5 KB)	100	0.98	0.90	0.99	**0.95**	14
------	------	------	------	------	------	------
GA(10 KB)	100	0.30	0.58	0.39	0.42	14
PSO(10 KB)	100	0.87	0.99	1	0.95	5
GA-PSO(10 KB)	100	0.99	0.91	1	**0.96**	17
------	------	------	------	------	------	------
GA(15 KB)	100	0.07	0.31	0.46	0.28	21
PSO(15 KB)	100	0.47	0.82	0.93	0.74	8
GA-PSO(15 KB)	100	0.91	0.99	0.94	**0.94**	32
------	------	------	------	------	------	------
GA(20 KB)	100	0.52	0.50	0.45	0.49	24
PSO(20 KB)	100	0.79	0.86	0.58	0.74	11
GA-PSO(20 KB)	100	0.99	0.91	0.91	**0.93**	55

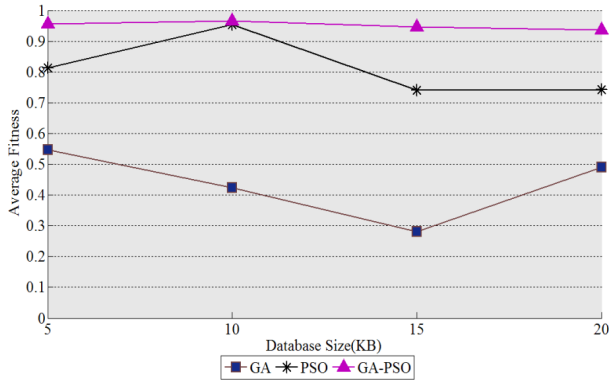


Fig. 5 The relationship between the Average Fitness and the database size

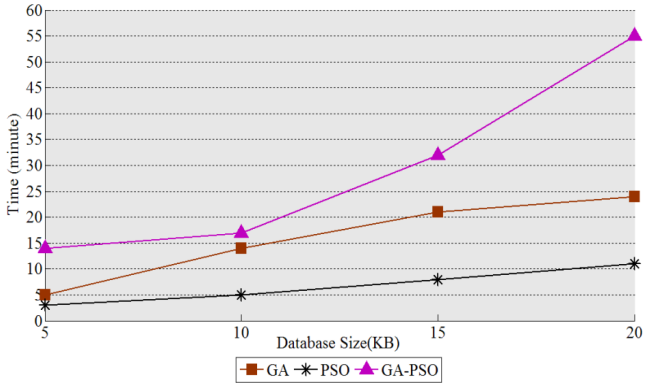


Fig. 6 The relationship between the execution time and the database size

5 Conclusions

In this paper, we have proposed a GA-PSO fuzzy data mining algorithm for extracting both association rules and membership functions from quantitative transactions. The experimental results have also shown that the proposed genetic-pso fuzzy mining algorithm have a good effect on fitness of membership function. In the future, we will continuously attempt to enhance the GA-PSO based mining framework for more complex problems.

6 References

[1] A.E. Eiben, M. Schoenauer," Evolutionary computing", Inform.Process. Lett. 82 (1) (2002) 1–6.

[2] A.E. Eiben, J.E. Smith, "Introduction to Evolutionary Computing", Springer, Berlin, 2003.

[3] J.H. Holland, "Adaptation in Natural and Artificial System", the University of Michigan Press, Ann Arbor, MI, 1975.

[4] D.E. Goldberg, "Genetic Algorithms in Search, Optimization &Machine Learning", Addison-Wesley, Reading, MA, 1989.

[5] Sushmita Mitra-Tinkuacharya "Data Mining Multimedia, Soft Computing, and Bioinformatics" - 2003 by John Wiley & Sons, Inc. (page64 & 267)

[6] Olivia Parr Rud – "Data Mining Cookbook Modeling Data for Marketing, Risk, and Customer Relationship Management"- 2001 - page311.

[7] Tzung-Pei Hong, Chun-Hao Chen, Yeong-Chyi Lee, and Yu-Lung Wu-"Genetic-Fuzzy Data Mining With Divide-and-Conquer Strategy", IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 12, NO. 2, APRIL 2008

[8] Matthew Settles, Terence Soule -"Breeding Swarms A GA-PSO Hybrid", GECCO, ACM 2005, 1-59593-010

[9] R. Eberhart and Y. Shi. "Comparison between genetic algorithms and particle swarm optimization". In e. a. V. William Porto, editor, Evolutionary Programming, volume 1447 of Lecture Notes in Computer Science, pages 611-616. Springer, 1998

[10] P. Angeline. "Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences". In V. W. Porto and et al., editors, Evolutionary Programming, volume 1447 of Lecture Notes in Computer Science, pages 601-610. Springer, 1998

[11] A. A. A. Esmin,G. Lambert-Torres,G. B. Alvarenga – "Hybrid Evolutionary Algorithm Based on PSO and GA mutation" Proceedings of the Sixth International Conference on Hybrid Intelligent Systems (HIS'06) 0-7695-2662-4/06 2006 IEEE

[12] X.H. Shi, Y.C. Liang, H.P. Lee, C. Lu, L.M. Wang - "An improved GA and a novel PSO-GA-based hybrid algorithm" Information Processing Letters 93 (2005) 255–261

[13] O. Cordón, F. Herrera, and P. Villar, "Generating the knowledge base of a fuzzy rule-based system by the genetic learning of the data base," IEEE Tran. Fuzzy Systems, vol. 9, no. 4, 2001

[14] A. Parodi and P. Bonelli, "A new approach of fuzzy classifier systems," in Proc. 5th Int. Conf. Genetic Algorithms, 1993, pp. 223–230

[15] C. H. Wang, T. P. Hong, and S. S. Tseng, "Integrating fuzzy knowledge by genetic algorithms," IEEE Trans. Evol. Comput. vol. 2, no. 4, pp. 138–149, 1998

[16] C. H. Wang, T. P. Hong, and S. S. Tseng, "Integrating membership functions and fuzzy rule sets from multiple knowledge sources, "Fuzzy Sets Syst., vol. 112, pp. 141–154, 2000

[17] F. Herrera, M. Lozano, and J. L. Verdegay, "Fuzzy connectives based crossover operators to model genetic algorithms population diversity," Fuzzy Sets Syst., vol. 92, no. 1, pp. 21–30, 1997

[18] M. Srinivas and L. M. Patnaik, "Genetic algorithms: A survey," Computer, vol. 27, no. 6, pp. 17–26, 1994

A Fuzzy Causal Decision-Making Model for IT Investment Evaluation

Pei-Chi Chen[1], **Ching-Chin Chern** [1], **and Gwo-Hshiung Tzeng** [2,3]
[1]Department of Information Management, National Taiwan University, Taipei, Taiwan
[2]Institute of Project Management, Kainan University, Taoyuan, Taiwan
[3]Institute of Management of Technology, National Chiao Tung University, Hsinchu, Taiwan

Abstract- *The decision-making process in information technology (IT) investment evaluation usually involves vague and imprecise human judgment. Our study proposes a causal decision-making model for understanding the influence relationship among evaluation dimensions and criteria by examining the evaluation process. Specifically, we employ the Fuzzy Decision Making Trail and Evaluation Laboratory (DEMATEL) technique to deal with fuzzy judgment. This technique demonstrates a comprehensive relationship map that allows for understanding the cause and effect relationship among evaluation dimension/criteria. The visualizing map also provides useful insights to decision makers. An empirical case in an automobile manufacturing company is used to exemplify this approach.*

Keywords: Fuzzy DEMATEL, IT Investment Evaluation, Causal Relationship, Decision-making

1 INTRODUCTION

In seeking of competitive advantage [1], enterprise today allocate increasing amount of their budget [2] in information technology (IT) investments. The decision-making process for IT investment evaluation is therefore getting important. However, during the process, enterprises usually need to consider several criteria [3] in order to make the best decision for the benefit of the company. A research conducted by [4] identifies twenty-three criteria such as price, quality delivery, etc. be considered for IT investment evaluation.

There usually exists some sort of relationship among evaluation criteria and oftentimes criteria may conflict with each other. The disagreement of evaluation criteria may affect decision-making results. Though an extensive number of decision models have been developed based on the multiple criteria decision making (MCDM) theory, such as analytical hierarchy process (AHP) [5], discrete choice analysis (DCA) [6], data envelopment analysis (DEA) [7, 8] and total cost ownership (TCO) [9], they lack examining the cause and effect relationship among evaluation criteria.

Understanding the causal relationship among evaluation criteria in IT investment helps the decision-making process. The Decision Making Trail and Evaluation Laboratory (DEMATEL) technique [11] can help for gathering group knowledge to form a structural model and provide a map for visualizing the causal relationship among constructs.

In addition, the evaluation for an appropriate vendor in IT investment usually involves vague or imprecise judgment during the MCDM process. In order to handle the vagueness in information and the fuzziness of human judgment or preference, Zadeh propose fuzzy set theory [12] in 1965 and a decision-making method in a fuzzy environment is developed in [13]. A number of subsequent studies used fuzzy set theory to deal with uncertainty in the vendor selection problem [14]. Due to the subjectivity of human judgment, IT investment is full of uncertainty and imprecision. Fuzzy set theory can be useful for dealing with this situation. In this study, we suggest fuzzy set theory as a way to improve the vendor selection problem.

Therefore, the DEMATEL technique is extended to combine fuzzy logic to help for IT investment decision-making in the fuzzy environment. This study attempts to exam the group decision-making process and proposes a causal model when selecting investment solutions in an enterprise. Specifically, this study aims to apply fuzzy DEMATEL technique to explore the causal relationship among evaluation criteria for better strategic IT investment.

The remainder of this paper is organized as follows. Section 2 presents the investment evaluation model that includes the construction of evaluation criteria, and the fuzzy DEMATEL technique. The evaluation model is further illustrated with a case study in Section 3. Section 4 discusses the research findings and its managerial implications. Finally, the conclusion is drawn in Section 5.

2 IT INVESTMENT EVALUATION MODEL

2.1. Hierarchical structure of evaluation criteria

The decision-making process for IT investment evaluation is usually complex and involves vague information. Oftentimes, organizations need to involve "several decision-makers" to face "several vendors and solutions" based on "numerous criteria". In such a context, MCDM theory can be used as an analytic

method to evaluate the advantages and disadvantages of alternatives based on multiple criteria [15].

In solving IT investment evaluation problem, several evaluation dimensions, which include several criteria, are used to help for decision-making. Through brainstorming sessions among management, comprehensive consultation with several experts, and extensive literature review [1, 5, 16, 17, 18], key dimensions of the criteria and criteria for IT investment evaluation were derived. The hierarchical structure of criteria in this study is shown in Table 1. Four dimensions which include Company's Internal Consideration, Company's External Consideration, Risk Consideration, and Consulting Company's Consideration are considered in this study. Among these four dimensions, 17 evaluation criteria for the hierarchical structure were used in this study.

Table 1 Dimension and Criteria of IT Investment Consideration in This Study

Dimensions	
D_1: Company's Internal Consideration	
D_2: Company's External Consideration	
D_3: Risk Consideration	
D_4: Consulting Company's Consideration	
	Criteria
D_1	Competitive advantage improvement (C_{11})
	Ease of Operation (C_{12})
	Information Quality improvement (C_{13})
	Meet user requirements (C_{14})
	Compatibility or integration ability with existing IT /IS portfolio (C_{15})
	Cost (hardware, and maintenance) (C_{16})
D_2	Adherance to government regulations (C_{21})
	Compatibility or integration ability with partners IS (C_{22})
	Ability to react to competitions promptly (C_{23})
D_3	Adequacy of manpower (C_{31})
	Skill of IT staffs (C_{32})
D_4	Consulting quality and obtainability (C_{41})
	Understanding level of consulting company to the organization (C_{42})
	Understanding level of consulting company to the industry (C_{43})
	Performance of previous similar projects (C_{44})
	Whether or not having good relationship with customers (C_{45})
	Cost (implementation cost) (C_{46})

2.2. Fuzzy DEMATEL technique

With different dimensions of criteria for IT investment evaluation, each criterion may impact the other criteria that affect the decision-making. Therefore, understand the causal relationship among dimensions/criteria helps the decision-making process. However, acquiring the knowledge of relationship among dimensions and criteria as mentioned above is complicated. It needs more flexible method to solve the problem. The DEMATEL approach was first adopted by [11, 19] to study the issues of huger, rave, environmental protection in the world. It was later applied in numerous disciplines such marketing [20, 10], education [22], agriculture [23], operation research [24], technology [25, 26], etc. This technique can verify the interdependency and reflect the evolution trend among criteria or variables [27, 28]. In addition, it allows for visualizing the structure of complex casual relationship by digraphs or matrices [10].

Owing to the reason that it is sometimes not easy for an evaluator to assess his/her opinion by a precise value, it would be better to use linguistic terms [12] such as high, medium, or low instead. A range value to represent linguistic variables can express opinions and feelings more accurately. Therefore, a fuzzy concept is introduced to represent the influence of each criterion to other criteria in our model. We use the linguistic variable, "influence" and range values in the proposed causal analysis model to overcome shortcomings of traditional DEMATEL approach.

2.2.1 The linguistic variable and its Triangular Fuzzy number for comparison

We first use linguistic variables, which are composed of words or sentences that can represent meaning in natural language, to get the relative influence of the criteria. Examples of linguistic terms used in this study are "No influence"(No), "Very low influence"(Vl), "Low influence"(Li), "High influence"(Hi) and "Very high influence" (Vh). The "no influence" means that cause criterion are perceived as "no influence" to the effect criterion while the "Very high influence" means the cause criterion is perceived as highly affect the effect criterion.

Then the triangular fuzzy numbers is used to enable the practicality of traditional DEMATEL technique. Simple operations such as addition, subtraction, multiplication and division of two fuzzy numbers [12, 29] are used to determine the membership functions. The triangular fuzzy number is denoted by (l_{ij}, m_{ij}, u_{ij}), where l and u stand for the lower and upper value of the support of a fuzzy number respectively, and the m represents the modal value. The mapping between linguistic terms and fuzzy level scale is developed by applying [21]. Table 2 shows the membership function of linguistic scale used in this study.

TABLE 2 Linguistic Scales For The Criterion Influence

Linguistic scale	Triangular fuzzy number
No influence (No)	(0, 0.1, 0.3)
Very low influence (Vl)	(0.1, 0.3, 0.5)
Low influence (Li)	(0.3, 0.5, 0.7)
High influence (Hi)	(0.5, 0.7, 0.9)
Very high influence (Vh)	(0.7, 0.9, 1)

2.2.2. Steps of Fuzzy DEMATEL technique

We summarize five steps to determine the evaluation criteria weights by Fuzzy DEMATEL as follows:

Step 1. Construct pair-wise causal relationship comparison matrices among all the criteria in the dimensions of the hierarchy system. Evaluators evaluate the influence they believe a criterion i impacts on every other criteria j, as indicated by $\tilde{a}_{ij}$. The direct-relation of criteria is thus represented by the matrix $\tilde{A}$. The

following matrix is an n× n matrix that defines the pair-wise causal comparison among criteria. For example, the first row of the matrix (1, $\tilde{a}_{12}...\tilde{a}_{1n}$) denotes the comparison between criterion 1 and the other n-1 criteria.

$$\tilde{A} = \begin{bmatrix} 1 & \tilde{a}_{12} & \cdots & \tilde{a}_{1n} \\ \tilde{a}_{21} & 1 & \cdots & \tilde{a}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{a}_{n1} & \tilde{a}_{n2} & \cdots & 1 \end{bmatrix} = [\tilde{a}_{ij}]_{n\times n} \quad (1)$$

Where $\tilde{a}_{ij} = (a_{ij}^{l}, a_{ij}^{m}, a_{ij}^{u})$

Step 2. Normalize the fuzzy direct-influence matrix obtained in Step 1. The normalized fuzzy direct-influence matrix of matrix $\tilde{A}$ is obtained by the following equation.

$\tilde{N} = v\tilde{A}$
where

$$v = \max_{ij}\left\{\frac{1}{\max_i \sum_{j=1}^{n}\tilde{a}_{ij}}, \frac{1}{\max_j \sum_{i=1}^{n}\tilde{a}_{ij}}\right\},\ i,j \in \{1,2,\ldots,n\}. \quad (2)$$

Therefore, the normalized fuzzy direct-relation matrix, $\tilde{N}$ can be represented as follows.

$$\tilde{N} = [\tilde{b}_{ij}]_{n\times n} = \begin{bmatrix} \tilde{b}_{11} & \tilde{b}_{12} & \cdots & \tilde{b}_{1n} \\ \tilde{b}_{21} & \tilde{b}_{22} & \cdots & \tilde{b}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{b}_{n1} & \tilde{b}_{n2} & \cdots & \tilde{b}_{nn} \end{bmatrix} \quad (3)$$

where $\tilde{b}_{ij} = v \times \tilde{a}_{ij} = \left(\tilde{b}_{ij}^{l}, \tilde{b}_{ij}^{m}, \tilde{b}_{ij}^{u}\right)$

Step 3. Calculate the fuzzy total-influence matrix. Once the normalized fuzzy direct-influence matrix is obtained, $\tilde{N} = (N^{l}, N^{m}, N^{u})$, the fuzzy total-influence matrix ***T*** can be calculated by applying the following equation.

$$\begin{aligned} \tilde{T} &= \tilde{N} + \tilde{N}^2 + \ldots + \tilde{N}^g \\ &= \tilde{N}(I + \tilde{N} + \tilde{N}^2 + \ldots + \tilde{N}^{g-1}) \\ &= \tilde{N}(I + \tilde{N} + \tilde{N}^2 + \ldots + \tilde{N}^{g-1})\left[(I-\tilde{N})(I-\tilde{N})^{-1}\right] \\ &= \tilde{N}(I - \tilde{N}^g)(I-\tilde{N})^{-1} \end{aligned} \quad (4)$$

where $\tilde{N} = [\tilde{b}_{ij}]_{n\times n}$

When $g \to \infty$, $\lim_{g\to\infty} \tilde{N}^g = [0]_{n\times n}$, then $\tilde{T} = \tilde{N}(I-\tilde{N})^{-1}$.

(6) And the fuzzy total-influence matrix ***T*** can be represented as,

$$\tilde{T} = [\tilde{t}_{ij}]_{n\times n} = \begin{bmatrix} \tilde{t}_{11} & \tilde{t}_{12} & \cdots & \tilde{t}_{1n} \\ \tilde{t}_{21} & \tilde{t}_{22} & \cdots & \tilde{t}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{t}_{n1} & \tilde{t}_{n2} & \cdots & \tilde{t}_{nn} \end{bmatrix} \quad (5)$$

where $\tilde{t}_{ij} = (\tilde{t}_{ij}^{i}, \tilde{t}_{ij}^{m}, \tilde{t}_{ij}^{u})$, and

$$\tilde{T}^{l} = \left[\tilde{t}_{ij}^{l}\right]_{n\times n} = \tilde{N}^{l}(I-\tilde{N}^{l})^{-1} \quad (6)$$

$$\tilde{T}^{m} = \left[\tilde{t}_{ij}^{m}\right]_{n\times n} = \tilde{N}^{m}(I-\tilde{N}^{m})^{-1} \quad (7)$$

$$\tilde{T}^{u} = \left[\tilde{t}_{ij}^{u}\right]_{n\times n} = \tilde{N}^{u}(I-\tilde{N}^{u})^{-1} \quad (8)$$

That means the triangular fuzzy numbers are divided into $\tilde{T}^{l}$, $\tilde{T}^{m}$, and $\tilde{T}^{u}$.

Step 4. Defuzzy the fuzzy values into crisp values. We use the method proposed by [30] to convert fuzzy data into crisp score. Therefore the fuzzy total-influence matrix are defuzzified into crisp value as the total-influence matrix,

$$T = [t_{ij}]_{n\times n} \quad (9)$$

Step 5. Construct the structural model and draw the causal diagram for analyzing the results. In total-influence matrix, the sum of the *i*th row represent the influence the criterion *i* affects the other *j* criteria both directly and indirectly; while the sum of the *j*th column represents the influence the criterion *j* receives from other *i* criteria. The sum of rows (denoted as r), and columns (denoted as c) in total-influence matrix, $T = [t_{ij}]_{n\times n}$ can be presented by the following equations,

$$r = (r_i)_{n\times 1} = \left[\sum_{j=1}^{n} t_{ij}\right]_{n\times 1} \quad (10)$$

$$c = (c_j)_{1\times n} = \left[\sum_{i=1}^{n} t_{ij}\right]_{1\times n} \quad (11)$$

where both r and c are vectors

In drawing the casual-relation map, the sum of vector r and c (*ie*, r+c) represents the horizontal axis. This is the total influence of each criterion. Similarly, the difference of vector r and c, (r-c), represents the vertical axis. This separates criteria into a cause group and an affected group. General speaking, the higher the value of r-c, the closer the criterion belong to the cause group. On the contrary, the lower the value of r-c, the closer the criterion belong to the affected group. The causal-relation map can be obtained by plotting the data set of $\{(r_i + c_i, r_i - c_i), i = 1,2,\ldots,n\}$. This map can provide visualized insight to the criteria relationship, enabling the decision-making in IT investment.

3 EMPIRICAL CASE STUDY: AN IT INITIATIVE IN AN AUTOMOBILE FIRM

The proposed causal initiative decision-making model based on fuzzy DEMATEL technique is applied to a company in Taiwan to demonstrate its ability for IT investment decision- making. The case company (referred to as CHMC) is a well-known automobile manufacturer in Taiwan. Starting with producing commercial vehicles, CHMC currently produce sedans, RVs, LCVs, and trucks.

Similar to the functions of other IT department, the IT department in CHMC endeavors to the computerization of the company. The IT department takes care of numerous systems, such as ERP, SCM, KM, DMS, CRM, etc. to help streamline the manufacturing process. Recently, CHMC has planned an IT initiative, which seeks to launch a Portal solution on the Internet. A variety of solution proposals such as IBM WebSphere solution, Oracle Portal solution or Microsoft SharePoint solution was proposed by the vendors. The policy in CHMC for IT investment requires the evaluation process for making final investment decision.

3.1 Multiple evaluation criteria

CHMC formed an evaluation team, which comprised of different stakeholders, including an executive manager, the IT manager, two IT staffs from IT division, and the manger and one staff from Purchasing department the team held several brainstorming sessions to review the evaluation criteria described in Section 2.1. Finally, the team members all agreed to keep all the criteria described in Section 2.1 for analyzing the investment decision. In order to obtain the causal relationship of dimensions/criteria and investment insights, fuzzy DEMATEL technique is applied in the case company.

3.2 Fuzzy DEMATEL technique applied in the case company

The pair-wise comparison questionnaire was designed based on the dimensions/criteria decided by the evaluation team. All the six survey were returned for the data analysis. The initial fuzzy direct-influence matrix $\tilde{A}$ was then constructed. The parameter v for normalization was calculated based on equation (3), v = 6.0253. The normalized fuzzy direct-influence matrix $\tilde{N}$ was also developed. By following equation (5), the fuzzy total-influence matrix $\tilde{T}$ was obtained. Due to the space limitation of the paper size, results of matrix $\tilde{A}$, $\tilde{N}$, and $\tilde{T}$ is not shown on this paper, but it will be presented in the conference.

The defuzzy technique as described in Step 4 of Section 2.2.2 was employed to obtain the crisp value of the total-influence matrix T. The result is shown on Table 3. The defuzzied influence of each criterion under each dimension is shown in Table 4. Finally, the influence relation map is drawn in Fig. 1 based on the data in Table 4.

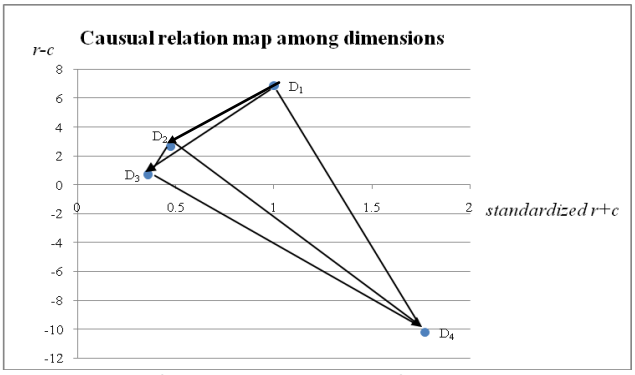


Fig. 1 Influence Relation Map for Dimensions

4 DISCUSSIONS AND IMPLICATION

Taking a close look at the causal relation map among dimensions, the results show that D_1 (Company's Internal Consideration) impact the other three dimension consideration, including D_2 (Company's External Consideration), D_3 (Risk Consideration) and D_4 (Consulting Company's Consideration). We find that the case company put much emphasize on internal consideration when making the IT investment decision. The results also show that the dimension D_2 impacts the consideration of D_3 and D_4, and the dimension D_3 impact the consideration D_4. This means that the consideration of consulting company is affected by the rest three dimensions in the case company.

In terms of criteria in dimension D_1, the results show that C_{12} (Ease of Operation) in is the causal consideration of the rest criteria in this dimension. When making IT investment decision in the case company, not only users but also the construction and maintenance party such as IT department put the operation easiness of system as the highest priority before considering other internal criteria such as C_{11} (Competitive advantage improvement), C_{13} (Information Quality improvement), C_{14} (Meet user requirements), C_{15} (Compatibility or integration ability with existing IT /IS portfolio), C_{16} (Cost on hardware and maintenance). One important finding to the case company is that C_{16} is affected by other criteria in dimension D_1, which in turn means that cost is the least important criteria in the internal consideration. This is attributed to enough budgets for the IT investment in the case company.

In terms of criteria in dimension of "Company's External Consideration", the results show that C_{21} (Adherence to government regulations) affects the rest criteria in this dimension. The case company requires strictly the obligation to regulations when investing IT solutions. Also, C_{23} (Ability to react competitions promptly) impacts the consideration of C_{22} (Compatibility or integration ability with partners IS). In terms of criteria in dimension of "Risk Consideration", the results show that C_{32} (Skill of IT staffs) affects the consideration of C_{31} (Adequacy of manpower).The case company focus more on the IT staff skills rather size of IT team.

Table 3 Defuzzied Total-influence Matrix T in The Case Company

	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	C_{21}	C_{22}	C_{23}	C_{31}	C_{32}	C_{41}	C_{42}	C_{43}	C_{44}	C_{45}	C_{46}
C_{11}	0.0234	0.0212	0.0537	0.0724	0.0446	0.0234	0.0332	0.0639	0.135	0.0312	0.0572	0.2234	0.2153	0.2532	0.2356	0.2245	0.2534
C_{12}	0.0521	0.0632	0.0645	0.0644	0.0425	0.0445	0.0162	0.0632	0.0325	0.0452	0.0246	0.2324	0.2452	0.2145	0.2452	0.2145	0.2542
C_{13}	0.0232	0.0245	0.0356	0.0225	0.0445	0.0245	0.0145	0.0532	0.0245	0.0145	0.0452	0.2312	0.2224	0.2245	0.2242	0.2532	0.2325
C_{14}	0.0346	0.0145	0.0352	0.0321	0.0378	0.0573	0.0142	0.0445	0.0352	0.0245	0.0242	0.2452	0.2245	0.2452	0.2246	0.2324	0.2452
C_{15}	0.0173	0.0145	0.0142	0.0115	0.0256	0.0246	0.0145	0.0412	0.0332	0.0325	0.0246	0.1245	0.2245	0.2246	0.2234	0.2312	0.2242
C_{16}	0.0324	0.0235	0.0332	0.0142	0.0245	0.1354	0.0332	0.0449	0.0224	0.0452	0.0573	0.1573	0.2324	0.2573	0.1421	0.2321	0.2234
C_{21}	0.0542	0.0323	0.0531	0.0434	0.0122	0.0425	0.0373	0.0421	0.0312	0.1242	0.0646	0.1246	0.1145	0.2452	0.1142	0.2521	0.2532
C_{22}	0.0231	0.0264	0.0325	0.0324	0.0252	0.0352	0.0446	0.0432	0.0352	0.1245	0.0532	0.1245	0.1245	0.1242	0.2325	0.2321	0.2134
C_{23}	0.0245	0.0473	0.0573	0.0512	0.0242	0.0342	0.0642	0.0645	0.0356	0.1241	0.0534	0.1234	0.1232	0.1242	0.1245	0.2242	0.2452
C_{31}	0.0525	0.0335	0.0546	0.0652	0.0345	0.0356	0.0625	0.0445	0.0354	0.1353	0.0512	0.1232	0.1421	0.1324	0.1523	0.2423	0.1134
C_{32}	0.0573	0.0421	0.0234	0.0532	0.0245	0.0452	0.0321	0.0445	0.0335	0.1145	0.0452	0.1241	0.1242	0.1142	0.1521	0.2452	0.1421
C_{41}	0.0246	0.0432	0.0321	0.0532	0.0245	0.0242	0.0324	0.0345	0.0242	0.1245	0.0242	0.1253	0.1356	0.1325	0.1321	0.1432	0.1256
C_{42}	0.0245	0.0235	0.0342	0.0132	0.0235	0.0632	0.0422	0.0453	0.0321	0.1021	0.0355	0.1432	0.1256	0.1534	0.1355	0.1532	0.1434
C_{43}	0.0245	0.0532	0.0645	0.0245	0.0321	0.0234	0.0432	0.0452	0.0421	0.1173	0.0242	0.1246	0.1245	0.1356	0.1234	0.1245	0.1684
C_{44}	0.0423	0.0323	0.0343	0.0235	0.0233	0.0534	0.0332	0.0442	0.0532	0.1242	0.0325	0.1246	0.1234	0.1422	0.1245	0.0126	0.1245
C_{45}	0.0632	0.0442	0.0445	0.0242	0.0321	0.0234	0.0232	0.0428	0.0631	0.1245	0.0126	0.1573	0.1421	0.1234	0.1521	0.1245	0.1242
C_{46}	0.0573	0.0142	0.0245	0.0132	0.0222	0.0452	0.0234	0.0342	0.0423	0.1221	0.0532	0.1246	0.1432	0.1325	0.1321	0.1421	0.1355

Table 4 The Defuzzied Influence of Each Criterion

Dimensions /Criteria	r_i	c_i	r_i+c_i	r_i-c_i
D_1 Company's Internal Consideration				
C_{11} Competitive advantage improvement	1.9646	0.631	2.5956	1.3336
C_{12} Ease of Operation	1.9189	0.5536	2.4725	1.3653
C_{13} Information Quality improvement	1.7147	0.6914	2.4061	1.0233
C_{14} Meet user requirements	1.7712	0.6143	2.3855	1.1569
C_{15} Compatibility or integration ability with existing IT /IS portfolio	1.5061	0.4978	2.0039	1.0083
C_{16} Cost (hardware, and maintenance)	1.7108	0.7352	2.446	0.9756
D_2 Company's External Consideration				
C_{21} Adherance to government regulations	1.6409	0.5641	2.205	1.0768
C_{22} Compatibility or integration ability with partners IS	1.5267	0.7959	2.3226	0.7308
C_{23} Ability to react to competitions promptly	1.5452	0.7107	2.2559	0.8345
D_3 Risk Consideration				
C_{31} Adequacy of manpower	1.5105	1.5304	3.0409	-0.0199
C_{32} Skill of IT staffs	1.4174	0.6829	2.1003	0.7345
D_4 Consulting Company's Consideration				
C_{41} Consulting quality and obtainability	1.2359	2.6334	3.8693	-1.3975
C_{42} Understanding level of consulting company to the organization	1.2936	2.7872	4.0808	-1.4936
C_{43} Understanding level of consulting company to the industry	1.2952	2.9791	4.2743	-1.6839
C_{44} Performance of previous similar projects	1.1482	2.8704	4.0186	-1.7222
C_{45} Whether or not having good relationship with customers	1.3214	3.2839	4.6053	-1.9625
C_{46} Cost (implementation cost)	1.2618	3.2218	4.4836	-1.96

In terms of criteria in dimension of "Consulting Company's Consideration", the results show that C_{41} (Consulting quality and obtainability) affects the rest criteria in D_4. The cause-and-effect sequence for the rest criteria is followed by C_{42} (Understanding level of consulting company to the organization), C_{43} (Understanding level of consulting company to the industry), C_{44} (Performance of previous similar projects), C_{46} (Implementation cost), and C_{45} (Whether or not having good relationship with customers). This shows that when choosing the consulting company for IT implementation, the case company focus mainly on consultants' ability to complete the IT project. Again, implementation cost is the least important consideration criteria, thus is affected by other criteria in this dimension.

5 CONCLUSION

IT investment in today's environment is inevitable and usually involves huge amount of capital budget and several stakeholders. Our purpose of this study was to develop a scientific framework for understanding the casual relationship among evaluation dimension/criteria for better decision- making. The DEMATEL technique serves as a useful approach to visualize the relationship in IT evaluation process. However, due to imprecise and vague information given by evaluator during the evaluation process, we propose fuzzy DEMATEL technique to compensate the weakness of traditional DEMATEL approach. This technique is further applied in an automobile manufacturer in Taiwan. The result of the study shows that the casual decision-making model that uses the fuzzy DEMATEL technique is an efficient and effective approach for casual analysis. The visualized relation map gives a clear understanding of the dimension /criteria relationship, thus helpful for decision-making in IT investment evaluation.

REFERENCE

[1] Powell, P.: Information technology evaluation: Is it different? *The Journal of the Operational Research Society*, Vol. 43, No. 1, pp. 29-42, 1992.

[2] Premkumar, G. and Ramamurthy, K.: The role of interorganizational and organizational factors on the

decision mode for adoption of interorganizational systems. *Decision Science,* Vol. 26, No. 3, pp. 303-336, 1995.
[3] Weber, C.A., Current, J.R. , Benton, W.C. :Vendor selection criteria and methods, European Journal of Operational Research, Vol.50,No. 1, pp. 2-18, 1991.
[4] Dickson, G.W.: An analysis of vendor selection systems and decisions, *Journal of Purchasing*, Vol. 2, No. 1, pp. 5-17, 1966.
[5] Ghoudsypour, S.H. and O'Brien, C.O.: A decision support system for supplier selection using an integrated analytic hierarchy process and linear programming, *International Journal of Production Economics*, Vol.56-57, No 1-3, pp. 199-212, 1998
[6] Verma, R. and Pullman, M.E: An analysis of the supplier selection process, *Omega*, Vol. 26, No. 6, pp.739-750, 1998.
[7] Weber, C.A. and Current, J.R.:A. Desai, An optimization approach to determining the number of vendors to employ, *Supply Chain Management: An International Journal*, Vol. 5, No. 2, pp. 90-98, 2000.
[8]Narasimhan, R. , Talluri, S. and Mendez, D.: Supplier evaluation and rationalization via data envelopment analysis: an empirical example, *Journal Supply Chain Management*, Vol. 37, No. 3, pp. 28-37, 2001.
[9] Degraeve, Z., Labro, E. and Roodhooft, F:An evaluation of supplier selection methods from a total cost ownership perspective, European Journal of Operational Research,Vol. 125, No. 1, pp 34-59, 2000.
[10] Wei, P. L., Huang, J.H., Tzeng, G.H., Wu, S.I.: Causal modeling of web-advertising effects by improving SEM based on DEMATEL technique, *Int. J. of Info. Technology & Decision Making*, Vol. 9, No. 5, pp 799-829, 2010.
[11] Fontela, E., Gabus, A.: The DEMATEL observer, DEMATEL 1976 Report, Battelle Geneva Research Center, Geneva, Switzerland, 1976.
[12] Zadeh L.A.: Fuzzy set. *Information and Control*, Vol. 8, No. 2, pp. 338-53, 1965.
[13] Bellman, R.E. and Zadeh, L.A.: Decision making in a fuzzy environment, *Management Science*, Vol. 17, No. 4, pp. 141-164, 1970.
[14] Boer, L., De, Labro, E. and Morlacchi, P.: A review of methods supporting supplier selection, *European Journal of Purchasing and Supply Management*, Vol. 7 , No. 2, pp. 75-89, 2001.
[15] Chou, T. Y., Chou, S. T., Tzeng, G. H.: Evaluating IT/IS investments: A fuzzy multi-criteria decision model approach, *European Journal of Operational Research*, Vol. 173, No. 3, pp. 1026-1046, 2006.
[16] Ryan, S.D., Harrison, D.A. Considering social subsystem costs and benefits in information technology investment decisions: A view from the field on anticipated payoffs. *Journal of Management Information Systems,Vol.* 16, No. 4, pp 11-40, 2000.
[17] Mirani, R. and Lederer, A.L.: An instrument for assessing the organizational benefits of IS projects. *Decision Science,* Vol. 29, No. 4, pp. 803-838, 1998.
[18] Jones, M.C., Beatty, R.C.: Towards the development of measures of perceived benefits and compatibility of EDI: A comparative assessment of competing first order factor models. *European Journal of Information Systems,* Vol. 7, No. 3, pp. 210-220, 1998.
[19] Gabus, A., Fontela, E. Perceptions of the world problematique: Communication procedure, communicating with those bearing collective responsibility. DEMATEL Report No. 1, Battelle Geneva Research Center, Geneva, Switzerland, 1973.
[20] Hsu, C.Y., Lee, W.H., Chen, K.T., Tzeng, G.H.: FMCDM with fuzzy DEMATEL approach for customers' choice behavior model. *International Journal of Fuzzy Systems*, Vol. 9, No. 4, 236-246, 2007.
[21] Mon D.L., Cheng C.H., Lin J.C. (1994) Evaluating weapon system using fuzzy analytic hierarchy process based on entropy weight. *Fuzzy Sets System*; 62 (2):127-34.
[22] Tzeng, G.H., Ou Yang, Y.P., Lin, C.T., Chen, C.B.: Hierarchical MADM with fuzzy integral for evaluating enterprise intranet web sites. *Information Sciences*, Vol. 169, No. 3-4, pp. 409-426, 2005.
[23] Kim, Y.: Study on impact mechanism for beef cattle farming and importance of evaluating agricultural information in Korea using DEMATEL, PCA and AHP. *Agricultural Information Research* Vol. 15, No. 3, pp 267-279, 2006.
[24] Ou Yang, Y.P., Shieh, H.M., Leu, J.D., Tzeng, G.H., A novel hybrid MCDM model combined with DEMATEL and ANP with applications. *International Journal of Operational Research*,Vol.5, No. 3, pp 160-168, 2008.
[25] Lee, Y.C., Li, M.L., Yen, T.M., Huang, T.H.: Analysis of adopting an integrated decision making trial and evaluation laboratory on a technology acceptance model. *Expert Systems with Applications*, Vol. 37, No. 2, pp1745-1754, 2010.
[26] Yamashina, H., Ishida, K., Mizuyama, H.: An innovative product development process for resolving fundamental conflicts. *Journal of the Japan Society for Precision Engineering*, Vol. 71, No. 2, pp 216-222, 2005.
[27] Chiu, Y.J., Chen, H.C., Tzeng, G.H., Shyu, J.Z.: Marketing strategy based on customer behaviour for the LCD-TV. *International Journal of Management and Decision Making*, Vol. 7, No 2/3, pp 143-165, 2006.
[28] Huang, C.Y., Tzeng, G.H. : Reconfiguring the innovation policy portfolios for Taiwan‟s SIP mall industry. *Technovation*, Vol. 27, No. 12, pp 744-765, 2007.
[29] Laarhoven PJM, Pedrycz W.: A fuzzy extension of Saaty's priority theory. *Fuzzy Set System*,Vol. 11, pp. 229-41, 1983.
[30] Opricovic, S., Tzeng, G.H.: Defuzzification within a multicriteria decision model. International Journal of Uncertainty, *Fuzziness and Knowledge-Based Systems*, Vol. 11, No. 5, pp 635-652, 2003.

A Fuzzy Inference System for Lightning Location

L. Barrera[1], S. Salazar[2], F. Bolaños[2]
[1]School of Mechatronics, Universidad Nacional de Colombia, Medellín, Colombia
[2]School of Mechatronics, Universidad Nacional de Colombia, Medellín, Colombia

Abstract – *Several techniques aimed to perform lightning location has been proposed on literature, most of them are based on using the time difference of arrival (TDOA) of electric signals to a set of sensors. These techniques often perform simplifications to the location problem, in order to deal with its complexity. This paper shows an approach based on the Takagi-Sugeno's fuzzy inference system [4], which can be trained for the sake to adapt it to the terrain conditions of the location system. The simulations show promising results, related both with the error of the location software as with its performance.*

Keywords: Location of electromagnetic sources, TDOA, Takagi–Sugeno inference systems.

1. Introduction

Systems for location of electromagnetic sources have been developed for applications such as Global Position Systems (GPS), radars and lightning's location schemes. Such systems often are composed by an array of sensors or detectors and a central processor for location's estimation.

For systems aimed to lightning's location, there are some drawbacks, such as the expensive installation, and the lack of adaptive features in order to tune the system to local conditions, where the sensors are placed. This is due to the complexity of the location problem and to the simplifications performed over it, such as assuming a constant propagation speed or a two-dimension scenario, without taking into account the local terrain's relief.

Several analytic strategies such as TDOA [5] and Range Difference have been proposed for radio sources location. TDOA seeks to estimate de source's coordinates starting from the differences of time measured among the sensors of the system. Range difference uses the range of distances between the sensors in order to infer the distance to the radio source.

Each approach implies the solution of a nonlinear system of equations, which means high complexity to the location system. There is also a major theoric constraint, since an isotropic medium is assumed for the terrain where the radio signal travels. This is not, of course, a real scenario, since the place where the system is placed has specific relief configurations. In order to overcome this problem, some range difference implementations are complemented with the use of error equations which are aimed to improve the quality of the estimation, since it considers the signal's delays.

Current approaches have to deal with a huge amount of calculations, in order to estimate the coordinates of a single lightning event. High performance computers are then required for implementing these systems. In some of these approaches, a coplanar setup of the sensors it is assumed in order to simplify the calculations. Some of these proposals, are the reported by Bard [1], Bucher [2] and Smith [3]. Due to its analytic nature, such solutions are not adaptive, and cannot be tuned to the specific terrain conditions where the system is located.

This paper is organized as follows. Section 2 describes the Takagi-Sugeno inference system. It shows the most relevant features of the implementation, as well as the training process in order to adapt the system to specific conditions of the terrain. Section 3 shows the implementation results of the inference system. Conclusions and future work are included in Section 4.

2. The Fuzzy Inference Approach

As many adaptive systems, the Takagi-Sugeno approach has two modes of operation. The training mode corresponds to the adjusting of some of the system's parameters; for the sake of adapt it to the specific problem at hand. The normal mode is such a regular operation state, in which the trained system responds as expected, within a margin of error.

Figure 1 shows a typical fuzzy inference system. Since fuzzy logic works with membership degrees, instead of numeric values, it is necessary to use a fuzzyfier at the input of the system and a defuzzyfier at its output [4]. The inference engine uses the membership degrees of the inputs and a set of logic rules, in order to calculate the output membership degree of the system.

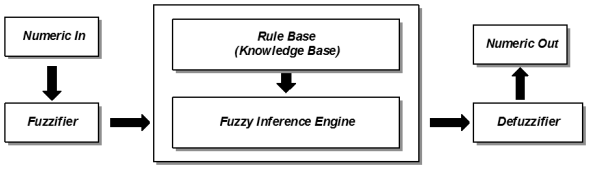


Figure 1. A typical fuzzy inference system.

In Takagi-Sugeno systems, activating a specific rule implies the use of a given linear combination of the input's membership degrees. Since more than one rule can

be activated at once, a weight factor is given to each linear function, in order to combine all the rules, by means of a sum.

The training process consists in finding the best values for the coefficients of the linear combinations associated to each rule, and for the weights given to each equation in the final sum. Adjusting these parameters is done by means of a minimal square reduction, starting from the error obtained at each iteration of the training algorithm. Figure 2 describes the training process graphically.

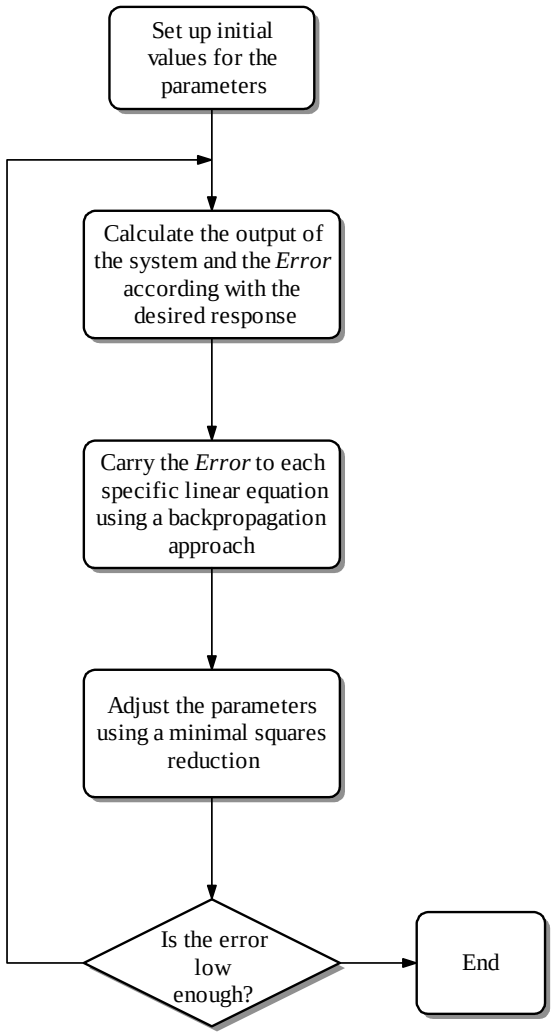


Figure 2. The training process.

The algorithm starts with setting up initial values for the parameters in the linear combinations and the weights of the inference engine. With these parameters, an output value is calculated, in order of obtain an error with respect to the desired value. This error is used to adjust the parameters by means a combination of a back propagation scheme and a minimal squares reduction. The adjusting of the parameters is performed iteratively until an error condition is reached.

The back propagation scheme works as follows: If F_i represents the output of each linear combination; the total output [4] of the inference engine (F) is calculated according with equation (1).

$$F = \sum_{i=1}^{M} F_i \times \beta_i \qquad (1)$$

Where β_i represents the weight given to each linear function and M is the amount of rules in the fuzzy system. When the output F is subtracted from the desired value, an error E is generated. The individual error (E_i) for each linear equation [4] is back propagated, according with equation (2).

$$E_i = \frac{E \times \beta_i}{\sum_{i=1}^{M} \beta_i} \qquad (2)$$

Once the individual error for each linear function is calculated, it is possible to adjust the parameters on such functions. Let's suppose that N is the amount of parameters for each linear equation, also suppose that $1 \leq j \leq N$. The new value of a given parameter (i.e. $A(i, j)_{NEW}$) of the ith equation, is calculated starting from its previous value ($A(i,j)_{OLD}$), according with equation (3).

$$A(i,j)_{NEW} = A(i,j)_{OLD} + LR \times E_i \times I_j \qquad (3)$$

Where LR represents the learning rate of the training process [4], and I_j represents the input of the given linear equation.

As can be seen, there are many things to tune in order to achieve a suitable performance of the Takagi-Sugeno inference system. The first thing to do is to determine the number of inputs of the system. In the case of lightning detection, it is necessary to count with at least three delays, in order to infer the location of the source without ambiguities.

Each of these inputs is normalized with values between 0 and 1, being 0 the maximum negative delay and 1 the maximum positive delay. This range is referred as Universe of Discourse. A partitioning process is performed in order to divide the universe of discourse on a set of fuzzy membership shapes. Figure 3 shows a universe of discourse for an input variable, which has been divided in three triangle-shaped membership functions.

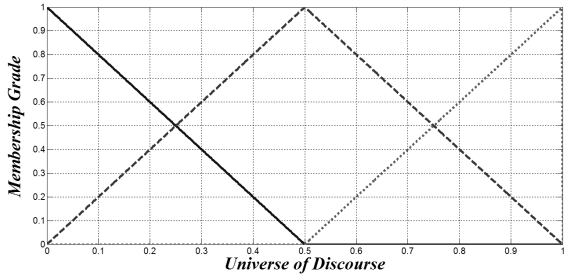


Figure 3. A given partition for an input variable.

If N is the number of input variables and S the amount of fuzzy membership functions for each input, the

number of fuzzy rules (M) can be calculated as shown on equation (4).

$$M = S^N \quad (4)$$

The last means that the amount of rules grows very quickly with the number of fuzzy sets used for the inputs of the system, so adjusting this parameter is a critical issue.

There is not a foolproof recipe with respect to the shape of the membership functions and the degree of overlapping among them. Instead, these issues are more related with the designer's experience and the amount of data available for training [6]. The only constraint related to the membership functions is that such functions must take always values between 0 and 1, with a continuous shape.

3. Obtained Results

The first parameter to consider in the development of this project is the coverage area of the antennas. For simulation and error estimation, a two dimensional terrain with an area of 200 km^2, has been considered. The system is aimed to detect the X and Y coordinates of single lightning events. The figure of merit being optimized is the mean error given by the system over a set of examples or training data.

In order to establish the right parameters to obtain the lowest possible error in the training process, several configurations of the fuzzy inference system was put under test. As mentioned before, the amount of membership functions for the input variables (namely N) must be determined, and also, there is a learning rate (LR) which must be tuned in order to achieve a good tradeoff between speed and suitable error values.

For instance, see case number 2 on Table 1, where each input variable was partitioned using five membership functions and a learning rate of 0.5 was used. The mean error in this case, resulted to be ± 1211 meters.

Table 1. Error varying the number of sets.

Case N°	N	LR	Error (meters)
1	3	0.5	2005
2	5	0.5	1211
3	7	0.5	684.8
4	8	0.5	819.6
5	10	0.5	956.5

The results show that there is an optimal value for the number of membership functions (N). The best error was achieved with N = 7 and resulted to be 684 meters (see case 3). Values of N below and beyond 7 show poorer results, in terms of the mean error in the location. This behavior could be explained as follows: low values of N implies coarse partition of the terrain's space, which implies at turn high errors in location. Very high values of N implies that the training process must tune much more parameter values (see equation (4)), so a more intensive training is required in order to achieve similar mean errors.

Following a similar approach, Table 2 shows the results of varying the learning rate, letting constant the amount of membership functions. The comparison was made again in terms of the mean error (quality of the training), for a fixed amount of algorithm's iterations.

Table 2. Error varying the learning rate.

Case N°	N	LR	Error (meters)
1	5	0.3	1154
2	5	0.4	1080
3	5	0.5	1211
4	5	0.8	1316

As can be seen on Table 2, there is not remarkable difference between the four cases, since the LR parameter affects only the speed in which the parameters are changed. This trend can be seen also on Table 3, where the LR parameter was changed for N = 7. Again, the mean errors resulted to be quite similar, except for the case when LR = 0.1.

Table 3. Error varying the learning rate and N = 7

Case N°	N	LR	Error (meters)
1	7	0.1	1101
2	7	0.3	716.6
3	7	0.5	684.8
4	7	0.6	613.2
5	7	0.7	584
6	7	0.8	655
7	7	0.9	646.9

Table 3 shows the best parameter tuning found in the simulations, which occurs with N = 7 and LR = 0.7. In this case, mean error was 584 meters.

With these defined parameters for the best case, it will be made below a fuzzy sets overlap analysis and how this affects the error training of the inference system. For this, was observed again the error results for overlap values lower than the maxim possible (0.5) because otherwise is not possible a convergence in the training.

Given the parameters already tuned (i.e. N and LR), the last tuning question is related with the overlapping between membership functions. When the triangle shape of a given membership function, overlaps 50 % of its neighbor function, it is said that overlapping is equal to 0.5. That's the situation depicted on Figure 2. Table 4 shows the error obtained for different values of the overlapping, letting N = 7 and LR = 0.7, since these are the best parameter values found.

Table 4. Error varying the overlapping for N = 7 and LR = 0.7

Case No.	Overlapping	Error (meters)
1	0.2	7757
2	0.4	1849
3	0.5	584

As can be seen on Table 4, the reduction of the overlapping between the fuzzy sets increases the training error. Therefore, it is suitable keeping the maximum value of overlapping between these sets.

4. Conclusions and future work

We have proposed and tuned a Takagi-Sugeno algorithm for lightning detection, with promising results, concerning the mean error of detection.

We have observed the effect of changing several settings of the algorithm, such as the learning rate, the number of membership functions and the overlapping among them. The evaluation of such settings combinations, have been performed in terms of the mean error achieved on the training process.

The obtained results have shown that changes in the overlapping of the fuzzy sets are not the best alternative in order to reach the minimal training error.

We propose as a future work, developing an algorithm that makes automatically the analysis and determination of the system parameters like learning rate, number of fuzzy sets and the fuzzy sets form in order to seek the minimal error. An adaptive clustering approach could help in achieving these goals.

5. Acknowledgments

The authors would like to thank to the National University of Colombia, for their support in this work's development.

6. References

[1] J. Bard, Fredric M. Ham and L. Jones, “An Algebraic Solution of the Time Difference of Arrival Equations” in Southeastcon '96. 'Bringing Together Education, Science and Technology’. Proceedings of the IEEE, pp. 313-319,1996.

[2] R. Bucher and D. Mira, “A Synthesizable Low Power VHDL Model of the Exact Solution of Three Dimensional Hyperbolic Positioning System” in VLSI Design, 2002 Vol. 15 (2), pp. 507–52, 2002.

[3] J. Smith and J. Abel, “Closed-Form Least-Squares Source Location Estimation from Range-Difference Measurements” in acoustics, speech and signal processing, ieee transactions on, ieee transactions on acoustics, speech, and signal processing, vol. assp-35, no. 12, December 1987.

[4] R. Babuska, “Fuzzy Modeling Control” in Kluwer Academic Publishers Norwell, MA, USA, 1998.

[5] B. Martin del Brío and A. Sanz Molina, “Redes Neuronales y sistemas borrosos”, 3° edición Alfaomega grupo editor SA, México, pp.440, 2007.

[6] H. Hellendoorn and D. Driankov, “Fuzzy Model Identification”, 1° edition, Springer-Verlag Berlin Heidelberg, 1997

SESSION

NEURAL NETWORKS + AGENT TECHNOLOGY + NLP + DATA MINING + ALGORITHMS AND NOVEL APPLICATIONS

Chair(s)

Prof. Hamid R. Arabnia

A Sales Forecasting Model for an Automotive Distributing Company

Jéssica Andressa de Souza[1], **and Celso Gonçalves Camilo-Junior**[2]
[1]Faculty of Sciences and Technology, Federal University of Grande Dourados, Dourados, MS, Brazil
[2]Institute of Informatics, Federal University of Goias, Goiania, GO, Brazil

Abstract - *One way to improve inventory management is to analyze sales forecasting information because those data can improve decision making. Thus, this paper proposes the application of Artificial Neural Networks (ANN) to generate two prediction models. The first one is for a product and the second model is for a group of products of an automotive distributing company in Brazil. Different architectures of ANN were tested to identify the best configuration. The results show a good performance of ANN for the scenarios analyzed.*

Keywords: Artificial Neural Networks, Forecast, Multilayer Perceptron, Data Mining.

1 Introduction

Data Mining (DM) is an area of computing that allows the work in the discovery of new information based on standards or rules with a large volume of data [14]. The terms Data Mining and Knowledge Discovery in Databases – KDD are commonly mistaken as synonymous. However, DM is a part of the KDD process [4], [19].

Among the various targets of DM and KDD, the prediction analyzes how certain variable will behave by other related attributes [5]. The forecasting is not an end in itself, but a means of providing information and support for a subsequent decision, aimed at achieving certain goals [12].

One application is the prediction of time-series forecasting, which is to provide the series value at time $t + h$ from a predictive model built based on a set of information from a series collected up to time t, illustrated in Figure (1).

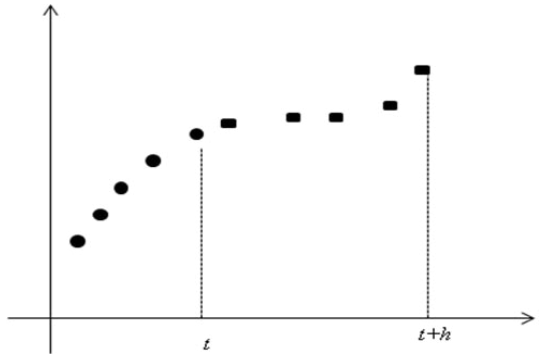


Figure 1 - Observations of a time-series forecasting horizon h and origin t. Source: [13].

The predictions of a time series can be classified as short, medium or long-term, depending on the forecast horizon. Thus, two different techniques to predict future values of a time series can be used [13].

- Forecast multi-step: where the set of current values is used to predict a given instant. The value of the next moment is determined from the current values plus the predictions already made. This technique is adopted for long forecast horizons as it seeks to identify trends and turning points in the most relevant series;
- Forecast simple step: where the prediction is made only for the time period immediately following the current, from observations of the current series

According to Souza (1989), ensuring the quality of forecasts of a time series is only achieved by adopting the forecast horizon as the instant of time immediately subsequent to the origin *t*.

There is a variety of methods of DM for pattern recognition, and some of these are of Artificial Intelligence. Among those, this paper uses the technique of Artificial Neural Network (ANN) as a tool to create a prediction model of a series. We opted for the ANN by good performance elsewhere. Comparisons with other methods of ANNs can be obtained by works [17], [11].

A good demand forecast means that stocks are better managed because from the moment the company becomes aware of the demand, it can prepare itself to make their decision-making [20].

Therefore, despite the importance of the prediction of sales in decision making, this paper discusses the use of ANNs to forecast demand in wholesale in an automotive distribution company. Among the various products available for sale, this work deals with the forecast of unit sales of the product 6167. which are considered one of the most financially important products for the company, and also the forecast of unit sales of a product group, which is this group's most financially profitable product as well.

This paper is organized as follows. Section 2 presents a brief summary of the Multilayer Neural Network (MLP) and the learning algorithm Rprop. Section 3 deals with time series forecasts. Section 4 presents the problem addressed in the workplace. Section 5 presents the experiments and Section 6 the results. Finally, Section 7 presents the conclusions.

2 Neural Networks For Multiple

Multilayer ANNs (Multilayer Perceptron - MLP) have the input layer, hidden layers and output layer [10] in their architecture. MLP ANNs have as main feature the ability to deal with issues not linearly separable, unlike the single-layer perceptron ANNs

The purpose of the hidden layers is to treat the non-linearity in order to reduce complexity for the output layer. For example, given a non-issue linearly separable, the middle tier of the ANN will make it linearly separable. The result is sent to the output layer so that it solves the problem given by the input layer.

MLP ANNs are progressive (feed-forward), i.e., the outputs of neurons in any particular layer are connected only to the entry of the next layer of neurons, without the presence of feedback loops. Consequently, the input signal propagates through the ANNs, layer by layer, in a forward direction [8]. Figure (2) illustrates an MLP ANN with two hidden layers .

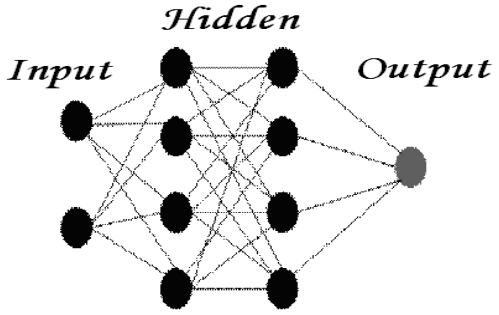


Figure 2 - ANN with an input layer (left) with two neurons, two hidden layers (center): the first with four neurons and the second with four neurons and an output layer (right) with one neuron. Source: [7].

2.1 Backpropagation

The Backpropagation is an algorithm that is based on supervised learning for correction of errors for training the MLP ANNAs [15]. Basically, learning consists of two steps: the spread, in which information - once processed - is propagated as the input layer to layer until the output layer, where the response is obtained from the ANN and the error is calculated, and the backpropagation where changes are made in the synaptic weights based on the output layer to input layer. The two steps are shown in Figure (3).

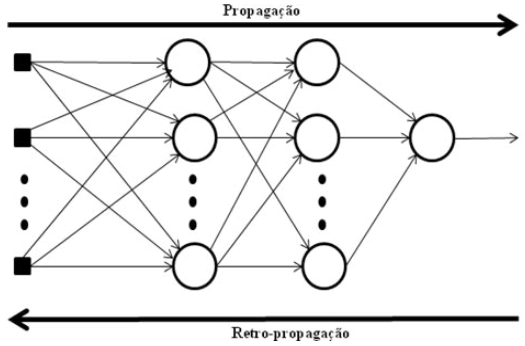


Figure 3 - Flow of the phases of propagation and back-propagation algorithm Backpropagation. Source: [1].

During the training phase, values of inputs must be present for ANN and also outputs desired values according to the inputs provided so as to allow the comparison of the ANN output with the desired output. Thus, the overall error of ANN that influences the correction of weights in the backpropagation step is estimated in order to decrease the calculated error [8, 3].

The aim of the Backpropagation algorithm, therefore, is on the surface of a global minimum error, i.e., values for the synaptic weights that minimize the error of the ANN.

The current value of synaptic weight is often overlooked. This does not only depend on the learning rate, but also on the partial derivative. The unpredictable behavior of the partial derivative affects the speed of the algorithm and its adaptability. This was one of the reasons that led to the development of Rprop.

To avoid the problem of adaptability, Rprop changes the size of the synaptic weights directly, i.e., without considering the size of the partial derivative [15]. These changes are intended to accelerate the training time of the algorithm and improve its performance since the standard Backpropagation algorithm is often slow.

2.2 Description do RProp

The algorithm Rprop (English Resilient Backpropagation) is a variation of the Backpropagation algorithm. It updates the weights and adjustments as soon as the gradient has computed all the standards and has made a direct adaptation of the weight value based on local gradient information.

The algorithm tries to eliminate the negative influence of the value of the derivative in the definition of partial adjustment of weights, as defined in Backpropagation. The algorithm eliminates the problem by leaving only the sign of the derivative, and not its value. Thus, the effort of adaptation is not overshadowed by the unpredictable behavior of the derivative [15], [3]

Take into consideration the example: when there is an increase in the error derivative and it is positive, then there is a decrease in the weight by the update value as indicated by Equation (1), otherwise the derivative is negative, then the update value is added according to Equation (2).

Equation (1)

$$[\![\Delta w_{ji}]\!](t) = \begin{cases} -\Delta_{ji}(t), \text{ se } \partial E/(\partial w_{ji})(t) > 0 \\ +\Delta_{ji}(t), \text{ se } \partial E/(\partial w_{ji})(t) < 0 \\ 0, \text{ se } \partial E/(\partial w_{ji}) = 0 \end{cases}$$

Equation (2)

$$\Delta_{ji}(t) = \begin{cases} n^{+}\Delta_{ji}(t\text{-}1), & se\ \partial E(t\text{-}1)/(\partial w_{ji}) \cdot \partial E(t)/(\partial w_{ji}) > 0 \\ n^{-}\Delta_{ji}(t\text{-}1), & se\ \partial E(t\text{-}1)/(\partial w_{ji}) \cdot \partial E(t)/(\partial w_{ji}) < 0 \\ \Delta_{ji}(t\text{-}1), & se\ \partial E/(\partial w_{ji}) = 0 \end{cases}$$

Under the rule of adaptation algorithm Rprop, if the partial derivative of error with respect to a weight w_{ij} keeps its signal, it means that its last adjustment made a reduction of the error, then the update value is increased by the object n^{+} leading to an increase in the speed of convergence of the training. But when the derivative changes its sign, this indicates that its last adjustment was too large, and then the update value is reduced by the object $\Delta_{ji}\, n^{-}$ changing direction of adjustment.

3 Estimated Time Series

A time series is a collection of observations made sequentially over time. The time series forecasts are forecasts of future values based on past values [6], [16].

A discrete time series can be represented by $X^T = (x_1, x_2, ..., x_T)$, where each discrete observation x_t is related to different sets, with a link of subordination between these observations [13]. The forecasts are not perfect and future involves uncertainty, but they provide information and provides help for an important decision-making in order to achieve certain objective.

3.1 Artificial Neural Networks in Time Series

ANN techniques have been applied to a variety of problem areas such as image processing, speech processing, forecasting optimization, among others, where you can get better results than conventional methods. However, understanding how the techniques of ANNs can achieve higher levels of operating performance continues to be difficult [18].

Among the problems mentioned above, time-series forecasting has received special attention from researchers. Predicting the future is fundamental in decision making, given the behavior of series that varies over time. This has been a great challenge for statistical and computing.

The ANN is presented as a great tool for forecasting time series. Its ability to extract difficult non-linear relationships from data entries that have noise has achieved remarkable results. In most cases, these have been even better than those achieved by conventional statistical procedures. Examples of applications of ANNs in forecasting time series are found in several applications: forecasting stock prices [13], prediction of drought [11]; forecast value of the commodity agribusiness, among others.

It is noted that statistical techniques do not provide good prediction results for some applications that have a set of sample size and restricted to some non-linearity in the data set [21]. Due to their great ability to learn, ANNs possess qualities that enable them to identify some characteristics of the series, such as timeliness, besides dealing with non-linear data [21].

Apart from their great advantages, the ANNs have disadvantages, such as identifying the best architecture of the ANN. That is, there is an appropriate setting for the application. The identification of architecture is discovered by trial and error. However, some heuristics suggested ranges of values.

4 Description of Problem

The problem, in this paper is based on the daily lives of a large company which has seven branches across the country with a turnover of millions per month. The company works with more than 20.000 different items from various brands and models. It has, therefore, a high complexity in the buying process, which is fundamental to good inventory management and enterprise resource.

To reduce complexity and improve the decision-making process of purchases, the company specialized and directed staff to certain brands of products, managing a division of tasks and, consequently, a reduction of complexity since each employee determines the purchase of a single brand.

One of the main characteristics of the company is its ability to prompt delivery of products with a gap in the market. This makes it one of the fastest selling in the sector. However, to achieve good financial results, it is necessary to estimate the size of the stock, which can be done with a good purchasing management.

In order to help the decision-making process, this paper addresses the projected amount of sales for a product brand and specific model and the amount of sales of that product group. To this end, it uses a database with 465 samples, 409 samples for training and 56 samples for testing/validation of ANN. Thus, the window of testing and validation of the week brought together the sales of two months.

The amount of sales of the product chosen for the tests was identified after preliminary analysis of data and conversations with employees, and the group that this product is bound varies with the shop (subsidiary) company and the week/month years. Therefore, we selected the information presented below for input ANN. Initially, there is a total of five attributes:

1. *Week*: variable identifying which week the sale was made, which ranges from 1 to 4;
2. *Month*: identification variable of the month the sale of the product took place, which ranges from 1 to 12;
3. *Seller*: identification variable branch. It varies from 0 to 6;
4. *QuantitySalesPreviousWeek*: provides the quantity of product sold in the previous week;
5. *Group*: Identifying which group the product belongs to.

It is believed that there were deals in the period analyzed that differentiated the sale of this product. As output of ANN was selected attribute *QuantitySalesWeek*, which provides an estimate of the total amount sold in the week.

6. After defining the attributes of entries, there was the process of standardizing the data from these attributes. This process meant that the data were normalized in a range of value ranging from -1.0 to 1.0. Data normalization allows better learning of the ANN.

5 Experiments

After pre-processing with the attributes, we sought to identify the amount of hidden layers and number of neurons in the hidden layers to the problem addressed. Finding a number of neurons for the layer(s) occult(s) is not an easy task because it depends on several factors, for example the amount of data available for testing and training, the quality of data available, among others [2].

If there are many neurons in layer(s) occult(s), the performance of ANN is suitable for the training data, but it tends to be bad for the test data/validation of the ANN, and it generates high computational cost. If there are few neurons of the ANN performance in layer(s), it may be bad both for training and for validation, despite the reduction in computational cost. That is, finding the perfect amount of hidden nodes is a costly task because the designer must always train and test several times to the ANN with different quantities of neurons and layers. The ideal number of neurons is one that can achieve the performance specifications suitable for both training data and for test data.

In order to identify the best architecture of the ANN, the tests were performed with one and two layer(s) occult(s) ranging from two to five neurons. This range was established by studying the rule Hayssler & Baum (1989). The rule proposes that the total number of parameters of the ANN (Z) calculated by Equation (3), and the amount of data available (N) obey a relation. See Equation (4).

Equation (3)

$$Z = (p+1)\, q_1 + (q_1+1)m$$

Where p is the dimension of input vector, m is the dimension of output vector EQA number of neurons in first hidden layer.

Equation (4)

$$N > Z / \varepsilon$$

Where ε is the error tolerated during testing, for example: if the error is = 0.1 (10% tolerance), this means that $N > 10Z$.

Among the various measures of accuracy, this paper adopts the Average Percentage Error (MSE), see Equation (5), the Mean Absolute Error (MAE), see Equation (6), and Mean Absolute Percentage Error (MAPE), see Equation (7).

Equation (5)

$$(\sum^{n}_{(i=1)} (((x_i - [\![x_i]\!]) / \hat{x}_i) (100)) / n$$

Where $\hat{x}_i$ is the observed value at time i;

x_i is the predicted value at time i and

n is the number of predictions made.

Equation (6)

$$(\sum^{n}_{(i=1)} (| ((x_i - [\![x_i]\!]) / \hat{x}_i |)) / n$$

Equation (7)

$$(\sum^{n}_{(i=1)} (| ((x_i - [\![x_i]\!]) / \hat{x}_i |)) / n\ (100)$$

The input data of the ANN were established through a program developed in Java, from data supplied by the company, filters and summarizes information. Thus, a new data table was generated and it is used for forecasting. This table was the entry for the software Knime [9], used for testing. To learn the ANN, we used the algorithm implemented in Rprop Knime. For each run 10.000 iterations are performed as preliminary tests indicate that value as the most appropriate.

6 Results

We applied the ANN with five neurons and one hidden layer Figure (4) and 2 hidden layers Figure (5) to forecast the number of units of a product group that was sold. We applied the ANN with five neurons and one hidden layer Figure (6) and 2 hidden layers Figure (7) to predict the amount of product units sold 6167.

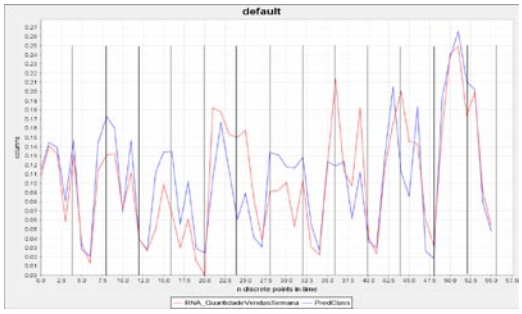


Figure 4 - Forecast for a group of products with the ANN of a hidden layer with five neurons.

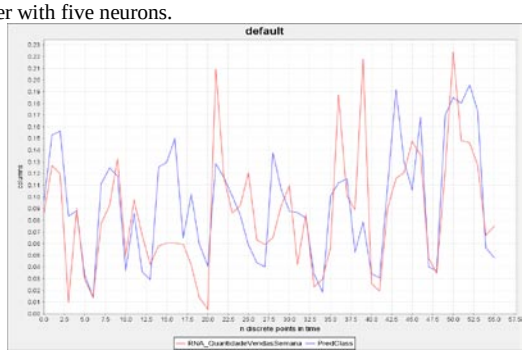


Figure 5 - Forecast for a product with ANN from a hidden layer with five neurons.

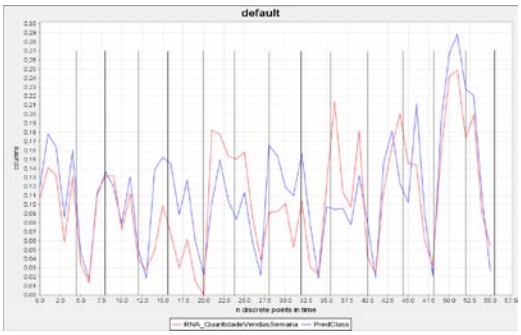


Figure 6 - Forecast for a group of products with the ANN of two hidden layers with five neurons.

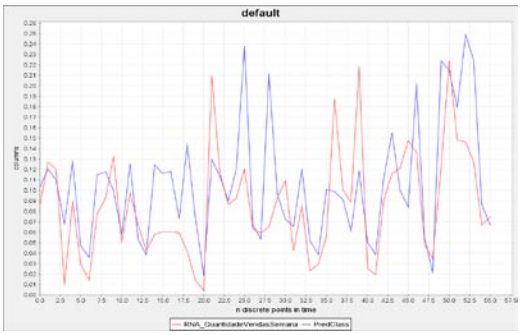


Figure 7 - Forecast for a product with ANN from two hidden layers with five neurons.

Some figures have vertical lines that delimit the series by store, so a more detailed assessment can be made. Store 0, store 1. store 2 store 3. store 4. 5 and shop shop 6 (series for the month of April) are analyzed from left to right and, after the end, a new count (series related to May) starts. The line has red corresponds to the desired value. The blue line is the predicted value by ANN.

Table 1 shows the results of AMI, the EPM and the MAPE of the forecasts for the group of each architecture.

Table 1 - Results of the forecast for the group.

Table of Forecast Errors for Group L = Layer; N = Neurons; E = Execution						
L	**N**	**E**	**EAM**	**EPM (%)**	**EPAM (%)**	**Error in quantity produtct (units)**
1	5	1	0,028	-2,722	2,8	
1	5	2	0,036	7,732	3,6	
1	5	3	0,036	0,035	3,6	
Average			0,033	1,682	3,333	234,353
1	4	1	0,036	1,103	3,6	
1	4	2	0,029	-2,709	2,9	
1	4	3	0,032	-1,082	3,2	
Average			0,032	-0,896	3,233	-124,840
1	3	1	0,035	1,327	3,5	
1	3	2	0,035	1,293	3,5	
1	3	3	0,032	3,779	3,2	
Average			0,034	2,133	3,4	297,191
1	2	1	0,036	1,299	3,6	
1	2	2	0,033	3,369	3,3	
1	2	3	0,029	7,96	2,9	
Average			0,033	4,209	3,267	586,486
2	5	1	0,038	-12,058	3,8	
2	5	2	0,034	5,268	3,4	
2	5	3	0,032	-1,299	3,2	
Average			0,035	-2,696	3,467	-375,680
2	4	1	0,036	1,042	3,6	
2	4	2	0,036	4,918	3,6	
2	4	3	0,037	1,999	3,7	
Average			0,036	2,653	3,633	369,642
2	3	1	0,035	1,313	3,5	
2	3	2	0,032	5,305	3,2	
2	3	3	0,036	-3,34	3,6	
Average			0,034	1,093	3,4333	152,241
2	2	1	0,04	2,93	4	
2	2	2	0,033	0,651	3,3	
2	2	3	0,035	5,906	3,5	
Average			0,036	3,162	3,6	440,608

Given these results, it is observed that the percentage index of the errors is small. Also, note that the results vary according to the store because its store time series is more complex than others. Observe that in stores 3. 4 and 5. ANN is more difficult since the stores 0, 1 and 2 the model is most effectively illustrated in Figures 5 and 6.

Whereas the results obtained are further points to the training points and the model takes into account the different stores, the ANN showed a good performance on average. For the worst case there was an EPM of -7.801% and in the best case of 1.422% with respect to the quantity of product units 6167 sold. Analyzing the error by the amount of plant products, 54.35 units in the worst case and 51.24 units at best.

Since the EPM, referring to the product group, presented the worst case 3.6% to 501.58 units of products of the group and presented the best case 3.3% to 390.12 units of the group's products. Thus, the difference in units between the worst and best case is 111.46 units.

It is noticed that the error both in predicting the product and in predicting the group would not cause considerable harm to the company since these values are small and easily absorbed in the following months.

7 Conclusion

The sales forecast is very important in making a business decision since it enables better management of inventory. That is, it helps in the purchase of quantity of products so that the

company does not lose sales and it does not increase the cost of inventory.

Therefore, this paper also forecasts sales of units of a product brand and model specific and group of products related to this product. It developed the model based on the time window of 2002 week by week from seven branches to predict the values of 2003 (April and May).

The results showed that it is possible to develop a predictive model based on ANN for the company analyzed. This model can predict wholesale sales of the product and the group that is satisfactory. Even with few input data, we obtained a model with good performance.

Given that there are several factors that affect the sales forecast, such as: promotions, commissions for salespeople, taxes, among others, we suggested as future work use more variable impact on the proposed model and an increase in the time series used in training. Moreover, we suggest using other ANN algorithms, e.g. static learning rate with Backpropagation [22].

8 References

[1] BARROS, Adélia Carolina de Andrade. "Otimização de Redes Neurais Artificiais para Previsão de Séries Temporais". Trabalho de Conclusão de Curso de Graduação (Engenharia da Computação), Escola Politécnica de Pernambuco. Maio, 2005.

[2] BAUM, E. B., & HAUSSLER, D. "What size net gives valid generalization?" Neural Computation, 6. 151–160, 1989.

[3] BRAGA, Antônio de Pádua; CARVALHO, André Ponce de Leon F. de Carvalho; LUDERMIR, Teresa Bernarda. "Redes Neurais Artificiais: Teoria e Aplicações". 2. Ed. Rio de Janeiro: LTC, 2007.

[4] CALIL, Leonardo Aparecido de Almeida; CARVALHO, Deborah Ribeiro; SANTOS, Celso Bilynkievycz; VAZ, Salete Marcon Gomes Vaz. "Mineração de Dados e Pós-Processamento em Padrões Descobertos". Publ. UEPG Ciências Exatas e da Terra, Ciências Agrárias e Engenharias, Ponta Grossa. Vol 14. Nº 03. p. 207 – 215 dez. 2003.

[5] FAYYAD, Usama; SHAPIRO, Gregory Piatetsky; SMYTH, Padhraic. "From Data Mining to Knowledge Discovery in Database". American Association for Artificial Intelligence. Providence, Rhode Island: FALL 1996. AAAI 97 - Conferences National de 1997.

[6] FIGUEREDO, José Clodoaldo. "Previsão de Séries Temporais Utilizando a Metodologia Box & Jenkins e Redes Neurais Artificiais para Inicialização de Planejamento e Controle de Produção". Curitiba. Dissertação (mestrado) - Universidade Federal do Paraná. 2008.

[7] FREIMAN, José Paulo; PAMPLONA, Edson de O. "Redes Neurais Artificiais na Previsão do Valor de Commodity do Agronegócio". V Encuentro Internacional de Finanzas. Santigo, Chile, 19 a 21 de janeiro, 2005. Disponível em: <www.iepg.unifei.edu.br/edson/download/ArtFreimanChile05.pdf> Acesso em: Junho, 2010.

[8] HAYKIN, Simon. "Redes Neurais Artificiais – Princípios e Prática". 2. Ed. São Paulo: Editora ARTMED, 1999.

[9] KNIME, Desktop. " Konstanz Information Miner". Disponível em: <http://www.knime.org/downloads-0> Acesso em Junho, 2010.

[10] KOVÁCS, Zsolt László. "Redes Neurais Artificiais: Fundamentos e Aplicações". 4. Ed. São Paulo: Editora Livraria da Física, 2006.

[11] MACIEL, Leandro S.; BALLINI, Rosangela. "Neural Networks Applied to Stock Market Forecasting: An Empirical Analysis". Journal of the Brazilian Neural Network Society, Vol. 8. Iss. 1. pp. 3-22. 2010.

[12] MORETTIN, Pedro Alberto; TOLOI, Clélia Maria de Castro. "Modelos para Previsão de Séries Temporais". In: 13º Colóquio Brasileiro de Matemática. Rio de Janeiro: (s.n), 1981.

[13] MUELLER, Alessandro. "Uma Aplicação de RNAs na Previsão do Mercado Acionário". Florianópolis. Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Departamento de Pós-Graduação em Engenharia de Produção, 1996.

[14] NAVATHE, Shamkant; ELMASRI, Ramez E. " Sistemas de Banco de Dados – Fundamentos e Aplicações". 4. Ed. Editora: Pearson Education (inglês), 2005. p. 622 – 643.

[15] RIDMILLER, Martin; BRAUN, Heinrich. "A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm". Proceedings of the IEEE International Conference on Neural Networks, San Francisco, CA, April, 1993. p. 586 – 591.

[16] SOUZA, Reinaldo Castro. "Modelos estruturais para previsão de séries temporais: abordagem clássica e Bayesiana", Instituto de Matemática Pura e Aplicada do CNPQ. IN: 17º Colóquio Brasileiro de Matemática. Rio de Janeiro, 1989.

[17] VIGLIONI, Giovanni Melo Carvalho. "Comparação entre Redes Neurais e Técnicas Clássicas para Previsão de Demanda de Transporte Ferroviário". Disponível em: <http://publique.rdc.puc-rio.br/rica/media/ICA01_Viglioni.pdf> Acesso em Junho, 2010.

[18] YOUNGOHC, Yoon; SWALES, George; MARGAVIO, Thomas. "Comparison of Discriminant Analysis versus Artificial Neural Networks. The Journal of the Operational Research Society", Vol. 44. Nº 1 (Jan., 1993), p. 51-60. Disponível em: <http://www.jstor.org/stable/2584434> Acesso em Junho, 2010.

[19] ZUCHINI, Márcio Henrique. "Aplicações de mapas auto-organizáveis em mineração de dados e recuperação de informação". Campinas. Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação, 2003.

[20] FLORES, João Henrique Ferreira; WERNER, Liane. "Aplicação de RNAs Neurais Artificiais à Previsão de Vendas de Máquinas Agrícolas". XXVII Encontro Nacional de Engenharia de Produção, Foz do Iguaçu, PR, Brasil. Out. 2007. Disponível em: <www.abepro.org.br/biblioteca/ENEGEP2007_TR620466_9360.pdf> Acesso em: abril, 2010.

[21] ABELÉM, Antônio Jorge Gomes. "Redes Neurais Artificiais na Previsão de Séries Temporais". Dissertação (mestrado), Rio de Janeiro – Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica. Setembro, 1994.

[22] CAMILO, Celso; YAMANAKA, Keiji. "A Practical Method for Finding an Efficient Static Learning Rate for ANN". Proceedings of the the 2008 International Conference on Artificial Intelligence, ICAI 2008, July 14-17, 2008.

Application of Neural Network to Detection of Cardiac Disease

Dr. Lakshmi Rajamani[1], Dr. A. Rajamani [2]
[1]Department of Computer Science and Engineering, Osmania University, Hyderabad-500007, INDIA.
[2] Department of Information Technology, Aurora Engineering College, Bhongir, 508116, INDIA.

Abstract - *This study is primarily aimed at the detection of various heart diseases of a human heart from the information available in electrocardiograph (ECG) signals using artificial neural networks. Data used is the disorders in ECG interpreted due to abnormalities in sinus rhythm, P wave, PR and QT intervals etc., Design and implementation of a neural network to diagnose heart diseases is dealt with in this paper. Seventy nine disorders in ECG are identified and these are used as input to the network. The network output has 36 nodes which correspond to various heart diseases. The network is trained using 180 patterns. The influence of learning rate, network architecture and momentum factor is studied. The accuracy in prediction is observed to be 85 percent*.

Keywords: Cardiac disease, neural networks, ECG

1 Introduction

An electrocardiograph is a non-linear signal that represents the electrical activity of the heart and describes the chronology of heart events. It corresponds to superposition of five different waves called arbitrarily P, QRS complex and T (Fig.1). Identification of the normal QRS complex from the P and T waves does not create difficulties as it has a characteristic waveform with dominating amplitude (~1 mv in a normal heart). The T wave follows the QRS complex after about 0.2 seconds. With advances in signal processing techniques, efficient data management and use of computers to process data faster and efficiently, computer medical systems are playing an important role in diagnostics. The earlier systems to interpret ECG waves employed decision tree classification rules for interpretation. However, this classification method is heavily dependent on the consistency of input data where measurements fall within or without certain ranges and Boolean combinations of each of these results determined whether the criteria for a certain diagnostic state were met.

In the recent past, artificial neural networks (ANNs) have provided an alternative means for medical diagnostics as they have high parallel information processing ability, self-learning, classification and associative memory [1,2,3,4,5]. Ning Ouyang et. al [2] diagnosed anterior wall myocardial infraction using 165 ECG records whereas Xie Zhi-xing et al [3] applied fuzzy network for diagnosing myocardial ischemia. Hu, Palreddy and Tompkins used an expert approach for the classification of ECS beats [4]. Wei Zang et al [5] classified ECG signals using block based neural networks. Further application of artificial neural networks for ECG analysis and arrhythmia detections are discussed in references [6,7]. In a recent publication, Dayang Gao et al [8] outlined a system for detection of cardiac arrhythmias within ECG signals based on a Bayesian ANN classifier. Its performance was evaluated by comparison with other classifiers. In this paper, ECG disorders and their correlation with various heart diseases are used in three layer architecture with 79 inputs (disorders) and 36 outputs (diseases). The influence of various parameters on the network performance is also studied.

2 Data Description

The data base used is analog clinic signals and it is presumed that these signals have been pre-processed [9] to get desired features to identify disorders in ECG. Twelve leads are generally used in the diagnostic ECG, although there is no limitation to the number of leads one may choose for special purposes. Leads I, II and III are all bipolar and detect an electrical potential change in the frontal plane. The other leads are aVR, aVL, aVF and V1, V2, V3, V4, V5, V6 [10]. The measures of ECG activity are P wave, QRS complex and T wave. From ECG signal processing, one can obtain amplitude, width, beat rate etc. in various leads and this data is used for disorders in ECG which become input to the network. ECG interpretation is done using the following information.

Inferior: Abnormalities that appear in leads II, III and F (called the inferior leads) indicate pathology on the interior or diaphragmatic surface of the heart.

Lateral: Leads I, F, V5-V6 are called lateral leads. Abnormality in these leads indicates pathology on the lateral upper surface of the heart.

Anterior: Anterior pathology is seen in leads V1-V3 and often in lead I.

Posterior: Problems in the posterior surface of the heart are difficult to diagnose using the standard 12 ECG leads. The pathology may be seen reflected through to V1 and V2.

Combination: Abnormalities may not be limited to one of the four areas described above. Inferolateral damage will show up in a combination of the inferior and lateral leads. Anterolateral damage will be seen in both anterior and lateral leads.

Various disorders (numbering 79) in ECG based on the information obtained from various leads are made. A few of these, for example, are: ST elevation in inferior leads, ST depression in anterior leads, wide QRS complex, tall R wave, invisible P waves, irregular rhythm etc. The disorders are used in detection of heart diseases which are categorized into ischemic heart disease, supraventricular rhythm, ventricular rhythm, hypertrophy patterns, atrioventricular (AV) block, etc. These have been further sub classified.

3 Artificial Neural Networks

Artificial neural networks emulate the basic characteristics of the human brain, such as massive parallelism and the capacity to process incomplete, unreliable and contradictory information without any claim to model the human brain in any way. Artificial neural net models have been studied for many years, especially in the fields of speech and image recognition. These models are composed of many non-linear computational elements operating in parallel and arranged in patterns reminiscent of biological nervous systems. Computational elements or nodes are connected via weights that are typically adapted during use to improve performance. Instead of capturing complicated processes in equations / formulae and then solving the problem analytically or numerically, neural networks can learn using only training data in the form of examples. In training, it is necessary to divide the data available for the application into a training data set, a validation data set and a test data set. A check is made during training to see whether the generalization error decreases monotonically with the number of training data passes or whether it reaches a minimum before rising again. They are of great use in machinery/medical diagnostics because of their ability to learn from measured data.

3.1 Back propagation network and training algorithm

Most of the successful demonstration of the potential of artificial neural networks used the back propagation (BP) algorithm. BP is a gradient procedure that first computes the gradient - with respect to network parameters (weight and thresholds) as a measure of the difference between the net's actual performance and its desired performance and then modifies the parameters accordingly. The most common measure of error is the mean squared error. Training a network comprises the following three steps [11].

- propagation of activation forward through the network
- propagation of errors backward, and
- modification of weights to reduce the error

Trained back propagation networks tend to give reasonable answers when presented with the inputs that they have never seen. This generalization property makes it possible to train a network on a representation set of input / target pairs and get good results. The back propagation network consists of one input layer, one output layer and one or more hidden layers. The number of inputs and the number of outputs are constrained by the problem whereas the number of hidden layers and the size of these layers (nodes) are up to the designer. The back propagation algorithm is an iterative gradient algorithm designed to minimize the mean square error between the actual output of a multi-layer feed-forward perceptron and the desired output. This can be expressed as

$$Ep = \tfrac{1}{2} \sum_{j} (Tpj - Ypj)2 \qquad (1)$$

where Ep is the measure of the error on input / output pattern 'p', Tpj is the desired output, Ypj is the actual output and let

$$E = \sum_{p} Ep \qquad (2)$$

be the overall measure of the error. The aim is to minimize equation (2). The various steps involved in the algorithm are described in [11]

4 Design and Implementation

The aim of any diagnostic scheme is to predict effectively normal/abnormal condition of a system from various symptoms observed. The heart disease can be detected by a cardiologist after a tedious process of analyzing various ECGs, comparison with previous history of the patient's records etc. Increased reliability and accuracy of the measurement systems, online data acquisition of data, signal processing, ability of computers to process data faster and efficiently, and networking can be used to develop a diagnostic system with cardiologist as a supervisor. The neural network model in the present case is designed and implemented for detection of heart diseases. There are no definite rules for setting of network design parameters as the solution space associated with each problem is not known apriori. Also, various network runs must be undertaken before the user can choose a suitable combination of

parameters with relative confidence. The various parameters are; number of input nodes, number of output nodes, number of hidden layers, number of nodes in each hidden layer, initial connection weights, initial biases, learning rate etc., Generally, the number of input and output nodes is fixed for a particular problem. The other parameters have to be selected by the user and tried for optimum results. The neural network model is implemented for detection of various heat diseases. There are 79 ECG disorders which can lead to 36 heart diseases. The modules created to implement the diagnostic scheme are:

Training Module: This module contains the input data for training, the number of layers, the size of each layer, and the values for the error tolerance and learning rate. The system begins training after the above information is fed. After the training is complete, the weight matrix is stored in a file to be used subsequently for testing. The output generated by the network is stored in an output file.

Interface Module: The interface module is used to provide the required interface to the program. This module has necessary instructions such as 'load data', 'save data', 'print data', 'learn from data', 'test data' to load the data file for testing purpose, saving the data after training, printing the saved data, to learn from loaded data and to test the provided data respectively.

Testing Module: In this module, the user provides the test data to the interface module which maps this data to the system in a file called 'test ECG.dat'. When this file is applied to a trained network, an 'output ECG.dat' file is generated.

Mapping Module: The aim of this module is to process the outputs generated during the testing phase and to display a more user friendly output in terms of diseases. The output generated during the testing module is simply either 0 or 1. This module maps the output from the test module with the associated disease and reports the same to the user.

Files used in various modules are TrainECG.dat, TestECG.dat, WeightsECG.dat, SaveECG.dat and OutputECGdis.dat. The class diagram in the network design is shown in Fig.2

180 patterns have been used to train the network having 79 inputs and 36 output nodes. A sample pattern is given below:

Input

0.97 0.95 0.6 0.65 0.92 0.95 0.65 0.1 0.1 0.1 0.26 0.28 0.05 0.06 0.12 0.16 0.1 0.1 0.26 0.28 0.1 0.26 0.28 0.05 0.06 0.12 0.16 0.52 0.05 0.06 0.12 0.16 0.05 0.06 0.12 0.16 0.05 0.06 0.12 0.16 0.05 0.06 0.12 0.16 0.05 0.06 0.12 0.16 0.05 0.06 0.12 0.16 0.05 0.06 0.12 0.16 0.05 0.06 0.12 0.16 0.1 0.1 0.1 0.26 0.28 0.1 0.26 0.28 0.05 0.06 0.12 0.16 0.52 0.05 0.06

Output:

1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

5 Results and Discussions

The application of a neural network for detection of heart diseases is considered. There are 79 inputs for ECG. A three layer architecture is selected based on author's experience on rotating machinery diagnostics [12]. The program is tested for various learning parameters for a fixed error tolerance. The optimal value of the learning parameter as observed in Table 1 is 0.9. A trial and error method was then tried to find the best architecture for given values of error tolerance and learning rate. Table 2 gives the results for a learning rate 0.9 and error tolerance of 0.01. Then the program is tested for various values of momentum factor (0 to 1) keeping the number of nodes in the hidden layer (45) and the learning rate (0.6) constant.

Table 1 Relation between learning rate and number of iterations

Learning Rate	Number of Iterations
0.3	58982
0.45	51436
0.5	48973
0.6	42651
0.75	34126
0.9	20298
1.0	26845
1.2	22545

The number of iterations is observed to decrease with momentum factor up to a value of 0.85 and is then found to increase again (Table 3). The network is tested with test patterns and observed to predict results with 85percent accuracy.

Table 2. Influence of Network Architecture on Number of Iterations.

Architecture	Number of Iterations
79-15-36	60454
79-25-36	49682
79-35-36	32256
79-40-36	25125
79-45-36	20298*
79-50-36	26956
79-60-36	31587
79-65-36	45687

Table 3 Relation between momentum factor and number of iterations.

Momentum Factor	Number of Iterations
0	43651
0.3	32784
0.6	18458
0.85	4872*
0.9	5121
1.0	7581

A typical test pattern has the following data:

Input

0.1 0.6 0.95 0.95 0.1 0.1 0.65 0.15 0.15 0.15 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.15 0.1 0.95 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.15 0.1 0.1 0.15 0.1 0.1 0.1 0.1 0.1 0.15 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1

Output

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0

The mapping module maps this data as the diseases ‘right bundle block’ and ‘sinus tachycardia’ into output ECG.dat

6 Conclusions

The artificial neural network is applied for detection of cardiac diseases based on the understanding of various disorders in ECG signals as obtained in various leads. A learning rate of 0.6 with a momentum factor of 0.85 can be used in a three layer architecture. The three layer architecture is observed to have an accuracy of 85 percent for test patterns.

7 References

[1] Baxt,W.G. Use of an artificial neural network for the diagnosis of myocardial infraction. ANN Intern. Med. 115, 843-848, 1991.

[2] Ning Ouyang, Mitsuru Ikeda and Kazunobu, Y. Using a neural network to diagnose anterior wall myocardial infraction, Proc. IJCNN’97, 59- 61, 1997.

[3] Xie Zhi-xing, Xie and Ning, X, Application of fuzzy neural network to ECG diagnosis. Proc. IJCNN, ’97, 62-66, 1997.

[4] Hu, Y.H, Palreddy,S and Tompkins, W.J.A patient adaptable ECG beat classifier using a mixture of expert approach. IEEE Trans.Biomed. Eng vol 44, 891-900, 1997.

[5] Wei Ziang, Kong, S.G and Peterson, G.D. ECG classification using block-based neural networks. Proc. IJCNN’05, 326-331, 2005.

[6] Silipo, R. d Merchesi, C.Artificial neural networks for automatic ECG analysis. IEEE Trans. Signal Processing, vol 46(5), 1417-1425, 1998

[7] Yang Wang et al. A short term multifractal approach for arrhythmia detection based on fuzzy neural network. IEEE Trans. Biomed Eng. Vol 48(9), 989-995, 2001.

[8] Dayong Gao, Michael, M., Des,C. and Gerard, L. Bayesian ANN classifier for ECG arrhythmia diagnostic study: a comparison study. Proc. IJCNN ’05, 2383-2388, 2005.

[9] Kunt, M., Rey.H. and Lightenberg, A Preprocessing of electrocardiograms by digital techniques. Signal Processing, vol 4, 1982

[10] Goldman, M.J. Principles of electrocardiography, (Maruzen asian edition), 1986.

[11] Simon Haykin. Neural networks – a comprehensive foundation. Second edition, Pearson Education Inc., 2001

[12] Lakshmi R. A neural network approach to fault diagnosis of pumps. Presented in AIPR'07, July 9-12, 2007, Orlando, Florida.

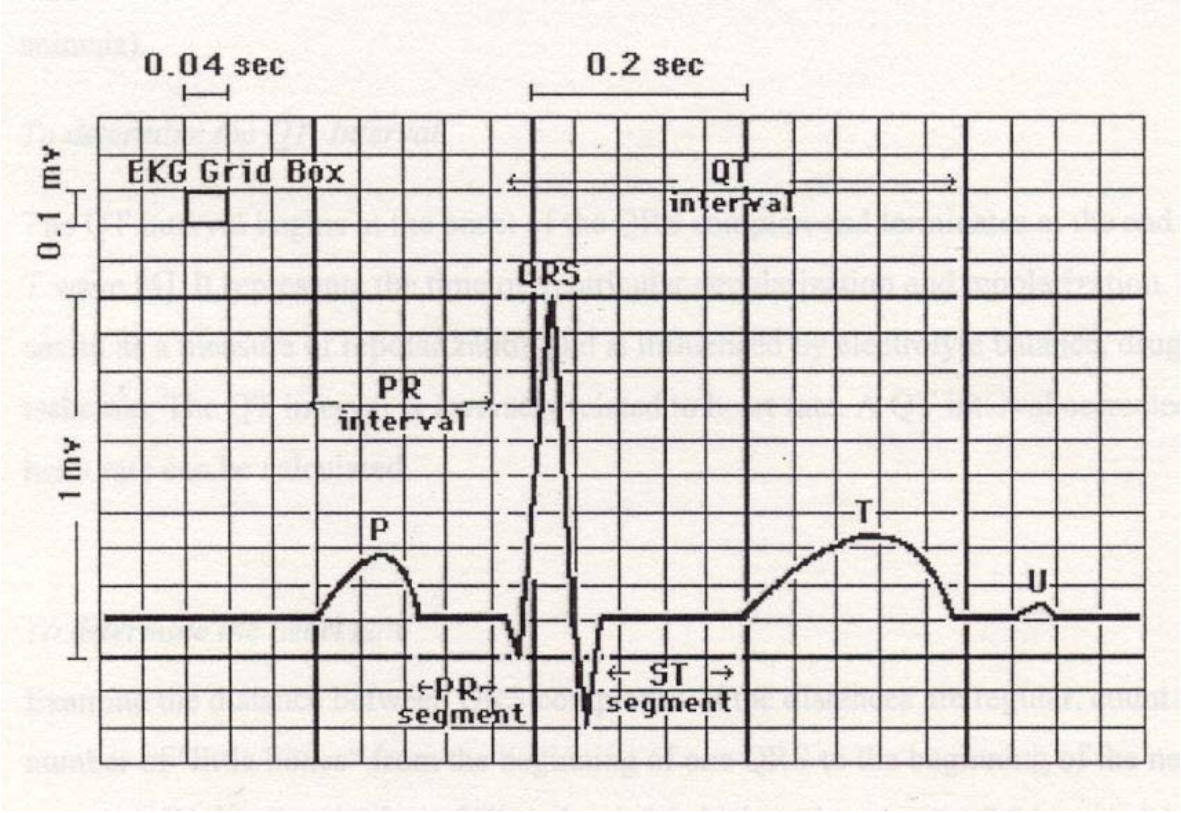


Fig 1. A typical ECG Signal

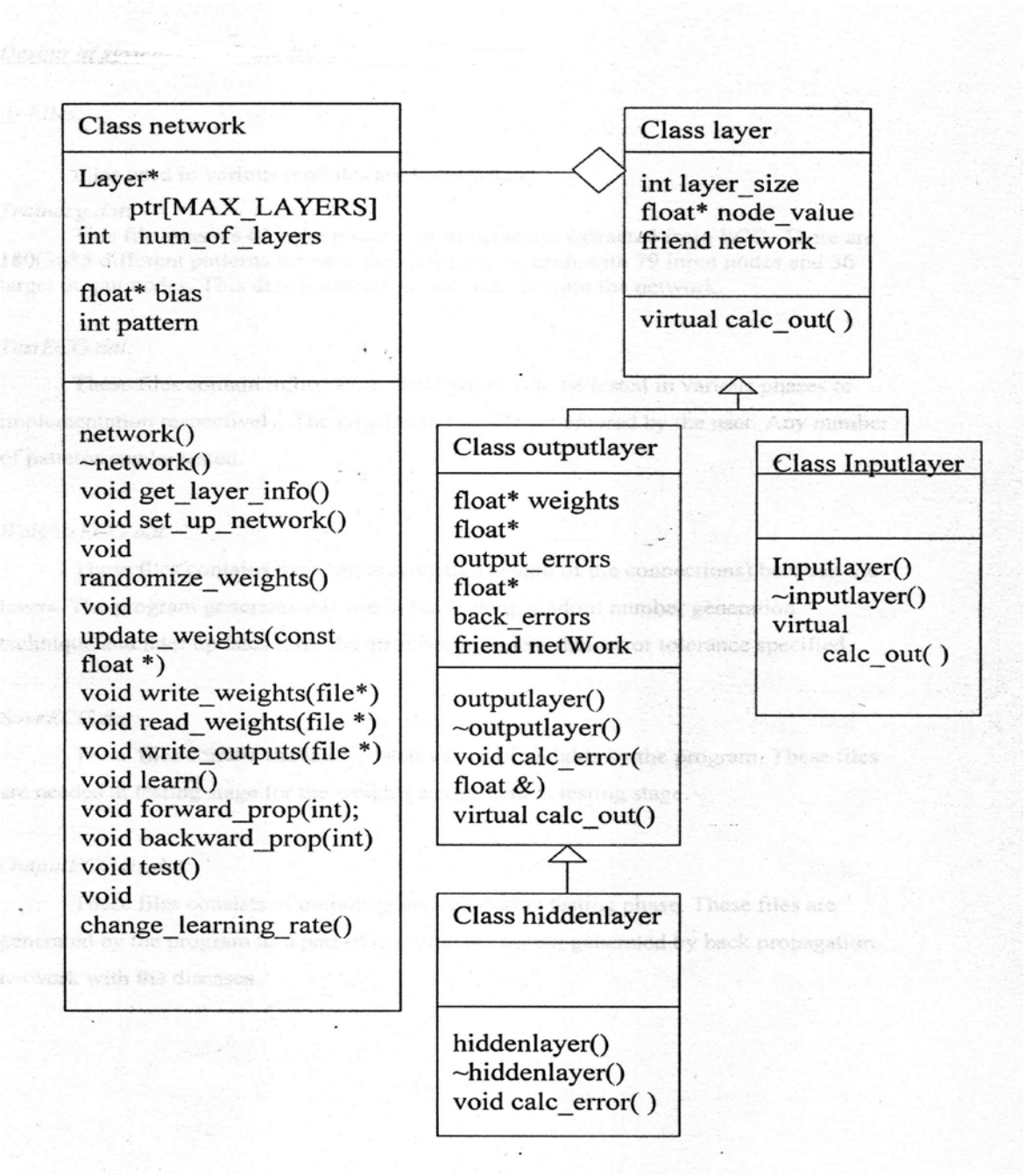


Figure 2 Class Diagram

A New Approach to Modeling Cognitive Information Learning Process using Neural Networks

Sajjad Mohsin, Fatima Zaka
Department of Computer Science
COMSATS Institute of Information Technology, Islamabad, PAKISTAN
smohsin@comsats.edu.pk,

Abstract

Recurrent Neural Networks have shown potential for cognitive modeling by showing good results with real-world temporal contextual data. Long Short Term Memory (LSTM) neural network model was designed to address the issue of large time lags in input data successfully. However, for the lower level cognitive processing, specifically, information processing, storage & recall, its performance had room for improvement mainly because LSTM cannot learn adaptively. Sustained Temporal Order Recurrent (STORE) networks are designed to encode the order of temporal data, and then could recall the encoded data in veridical as well as non-veridical order employing unsupervised learning. By the fusion of LSTM and STORE, a new neural network model LSTM-STORE is proposed that can alternate between supervised and unsupervised learning modes. The key purpose of this fusion is to mimic the working of brain for information processing during active as well as sleep time. To mediate the shift in the two learning modes we proposed Consolidation Control System CCS, which through gating mechanism controls the flow of information though the proposed network. The proposed model LSTM-STORE not only aims to investigate the working of LSTM in an unsupervised learning based environment; but also to perform lower level cognitive reasoning tasks. Thus LSTM-STORE basis it's plausibility on a comprehensive cognitive foundation. The proposed model is justified by providing comparative experimental proof.

1. Introduction

Brain can perform simple or intricate cognitive tasks by employing supervised as well as unsupervised learning. Also Researchers and cognitive psychologists stress on the presence of veridical as well as non-veridical ordered processing in the brain, enabling it to undertake various difficult to simulate tasks, as well as understanding various real-world phenomenon and experiences better. [1][13] Few examples exist of unsupervised learning with respect to temporal data and employing recurrent nets to model lower level cognitive reasoning processes. One example is a hybrid of recurrent neural networks (employing the extended kalman filters for training the recurrent net) with ART2. It was designed [17] to have the ability to adaptively learn in response to varying input patterns and then it further transfers this learning to dynamically growing groups of simple recurrent nets. Similarly Adaptive Resonance Theory (ART) neural networks have been used for specific applications dealing with temporal [18] as well as spatial-temporal input sequences [15]. Bradski et al proposed a variant of basic ART architecture that could store as well as recall various temporal input sequences. Sustained Temporal Order Recurrent (STORE) model although can not process the stored values to perform any tasks, but their key objective is to serve as working memory to other networks. [3]
Long Short Term Memory (LSTM) performs well where recurrent neural networks fail specially with respect to time lag problems, and performs better than other approaches to solve various bench mark problems. [10] But the gradient learning algorithm of LSTM cannot support unsupervised learning (error-based learning algorithm). In [11] LSTM was trained using two unsupervised learning algorithm to discriminate groups of temporal sequences. But no comparison with other benchmark solutions was provided in this work and application scope was limited. Furthermore, a preliminary critical analysis determined that the main focus of the application of LSTM for solving cognitive reasoning problems remains on high level reasoning problems. Problems like learning languages (Reber Grammar, Context Free & Context Sensitive) [6], speech recognition [8], and hand-writing recognition [7] were quite successfully handled by LSTM architecture, and exceeded in terms of performance and robustness while in comparison with other approaches. A system required for modeling cognitive information processing tasks should be able to accept and learn real world temporal input sequences, store this input sequence, recall this input sequence in any order (cognitive scientist predict non-veridical recall of information during processing [9]). And if any pre-learned information is presented to the system the system should be able to recognize it. This can only be

done if the proposed system has the ability to learn in supervised as well as unsupervised manner.

LSTM architecture can cater to lower-level reasoning tasks with ease as well if the architecture is modified to support unsupervised learning along with supervised learning. Also although LSTM can extract information conveyed by the temporal order of widely separated inputs but the basic architecture does not facilitate the non-veridical recall of any previously given temporal sequences.

In this paper a new approach is proposed to model the cognitive information processing and storage employing LSTM model. The proposed solution involves making certain additions to the basic LSTM architecture by fusing it with a working memory and the most suitable choice for working memory was STORE, as it caters to temporal data, supports unsupervised learning, and allows veridical as well and non-veridical recall of information.

Out line: The rest of the paper is organized as follows: section 2 reviews the biological plausibility giving the foundation for the proposed architecture. Section 3 and 4 gives briefly describes LSTM and STORE networks respectively; computational model as well as the working of the new architecture is given in section 5. The experimental setups and results are given in section 6, and section 7 concludes the paper with discussion on the proposed model and comparative results.

2. Cognitive Foundation

All the widely accepted theories of cognitive information processing like the "stage theory" [2] and the "level of processing theory" [5] explains that humans while learning and processing information make use of various distinct levels of elaboration. Then we have theories like parallel-distributed processing and connectionist [14] [12]: as their names indicate these theories give the idea that information is processed simultaneously but by several different parts located in our memory system. How the brain stores the data, and learns it in what order is still not known, the known part is that it stores and learns this data in an order that makes it's working efficient and robust. The brain learns new information and classifies input data based on target data provided by external teacher, this is supervised learning for brain. The brain by employing unsupervised learning can form perceptions based on previously learned concepts and information; and use the new data from sensory organs and these pre-formed perceptions to take decisions and perform tasks in an optimized manner.

According to above given arguments in order to suggest a system modeling cognitive information processing skills we need to benchmark certain qualities like: it can process in parallel, where tasks are divided in different units of processing; the system should definitely be able to cater to real world data (which is temporal in nature mostly); and this system should be able to encode the information as well as the order of input, and recall information even in non-veridical order. Similarly one cannot ignore the required element of unsupervised learning in the proposed system.

3. Long Short Term Memory (LSTM)

LSTM [10] forms a key part of proposed architecture; this section will give its architecture and key equations for the forward pass (activation) and backwards pass (gradient calculation) of the hidden layer in LSTM.

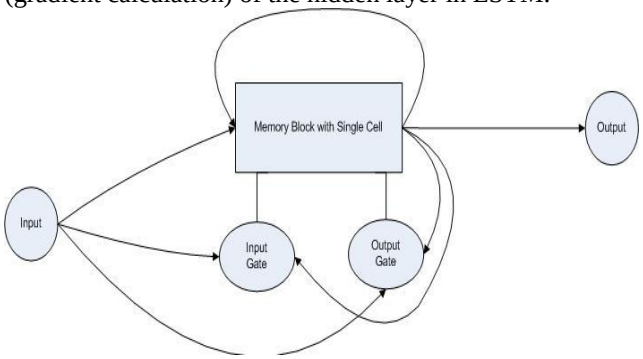


Figure 1: Long Short Term Memory

The hidden layer in a LSTM cell comprises of a memory block (having one or more memory cells) gated by two multiplicative gating units: input gate and an output gate (Fig. 1). Each memory cell contains a self-connected unit called "Constant Error Carousel" (CEC). The activation of CEC is generally termed as cell state. CEC solves the vanishing gradient problem by maintaining a constant flow of error during large intra-input time lags. This constant flow of error is maintained by Input and Output gate. The equations for forward pass and backward pass are:

Forward Pass:

For Input and Output gates:

$$net_{out}(t) = \sum w_{out-k} y^k(t-1); z^{out}(t) = f_{out}(net_{out}(t)) \quad (1)$$

$$net_{in}(t) = \sum w_{in-k} y^k(t-1); z^{in}(t) = f_{in}(net_{in}(t)) \quad (2)$$

where:

k = is the index ranging over input cells, as well as both the gates cells and also any other hidden units in the network

f = is sigmoid function here with range chosen as [0,1]

w = obviously indicates the weights between an arbitrary unit k and either input or output gate unit.

Similarly for the CEC we have the following equations:

$$net_{C_j} = \sum w_{c_j k} y^k (t-1) \quad (3)$$

Therefore the internal state of the CEC can be calculated as:

$$S_c(t) = S_c(t-1) y^{in}(t) g(net_c(t)) \quad (4)$$

LSTM also allows stopping the CEC from making some unwanted changes therefore the internal state Sc is squashed by using the squashing function h. Equation for that is:

$$y^c(t) = y^{out}(t) h(S_c(t)) \quad (5)$$

Backward Pass:

The backward pass encompasses the learning of the network by employing the truncated backpropagation through time [10]. The truncated version ensures that the internal state of CEC only learns/adapts when the error gets propagated via the input gate, any other random unit can not change the error flowing through the CEC, thus enabling it to hold on to its state for longer duration of time lags.

Similarly for backward pass the traditional equation comparing output and target holds true, this equation computes the mean-square error, and then this error is propagated back in time for weight updating:

$$E(t) = \sum_l (t^l(t) - y^l(t))^2 \quad (6)$$

where $t^l(t)$ is the target value.

The gradient-based update to weight between any two arbitrary unit l and m with respect to time is calculated as:

$$\Delta w_{lm}(t) = -\alpha \frac{\partial E(t)}{\partial w_{lm}} \quad (7)$$

Similarly any arbitrary unit l's error at time step t is given as[10]:

$$e_l(t) := -\frac{\partial E(t)}{\partial net_l(t)} \quad (8)$$

4. Sustained Temporal Order Recurrent (STORE)

STORE architecture is built on two principles: principle of invariance (which is used by the long term memory (LTM) to learn new patterns without forgetting old patterns), and normalization rule (STM activity value has an upper bound which it cannot exceed). The Standard working memory model proposed originally was STORE 1 [3]; it has two-layers as in ART, F1 and F2 (Fig. 2). The first layer (F1) is in fact a competitive system, and the activity vector (Xl, X2, ..., Xn) stored in this layer represents working memory. While the second layer (F2) tracks and stores the STM activity of the first layer via its activity vector (Yl, Y2,. .., Yn). The symbol of α is used to denote intra-input time lag, and β to represent inter-input time lags. F0 denote the input layer with values I1, I2… In

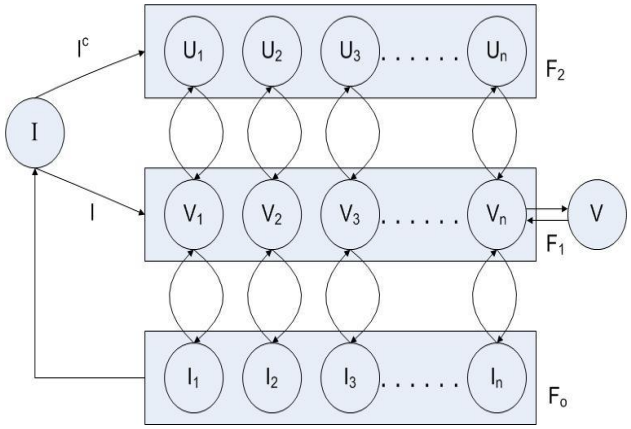


Figure 2: STORE 1 Model

At a time t when a unit input Ii is applied by F0 (the input layer), the first the input Ii is normalized in a winner take all manner setting it either to 0 or 1 accordingly,

$$I_i = \begin{cases} 1 & if\ \alpha - t_i < t < t_i \\ 0 & otherwise \end{cases} \quad (9)$$

With the presentation of new input pattern internal STM activation stored in F1 go through a gradient change which then accounts for the new input pattern.

Therefore at F1:

$$\frac{dx_i}{dt} = (AI_i + y_i - x_i x)I \quad (10)$$

where x=∑xk and I=∑Ik and A is invariance control parameter

At F2

$$\frac{dy_i}{dt} = (x_i - y_i)I^c \quad (11)$$

where Ic=1-I, Initially, xi(0)=yi(0)=0

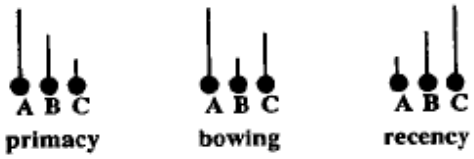


Figure 3: Invariance gradient [Bradski, Carpenter and Grossberg, 1994]

Tl l stores an input pattern, this property shows the invariance. The order of input pattern stored can be primacy, bowing and recency (see Fig. 3), and it is chosen with the variable A; i.e. if A is small (very small) the F1 stores primacy gradient, if 0<A<1 then bowing can occur, and if A≥1

then F1 stores a recency gradient. Temporal order of recalled information with respect to activity pattern stored across F1:

- Primacy: zk-1(ti) > zk(ti)
- Recency: zk(ti) > zk-1(ti)
- Bowing (combination of both primacy and recency)

One property of STORE network which is again taken from human working memory is the normalization property which indicates the maximum total activation F1 can store.

As for output: the most active node as shown in the Fig. 2 reaches its output threshold first, and is recalled or transmitted; afterwards this node self inhibits its activation via a negative feedback pathway therefore enabling the next most active node to be recalled. [3]

5. LSTM-STORE: Architecture and Computational Model

A new architecture LSTM-STORE is proposed by fusion of LSTM and STORE for modeling the lower lever cognitive information processing. The design of the new architecture draws its inspiration from the cognitive foundation provided in section II. The Architecture is given Fig. 4.

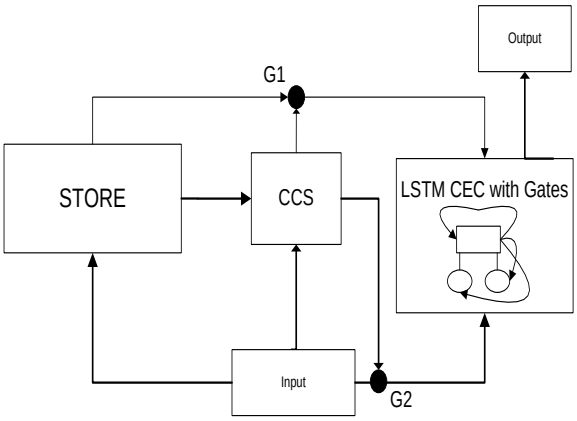


Figure 4: LSTM-STORE Architecture

LSTM-STORE gives a solution with levels of processing and also where processing is being performed in parallel, in distinctive areas of network. The distinct areas of parallel processing (as shown in Fig. 4) are the LSTM and STORE, and the Consolidation Control System (CCS) unit enables the proposed model to alternates between supervised and unsupervised processing. The temporal input (real world data with time lags) is simultaneously fed to the STORE and LSTM units; while LSTM learns to classify the sequence, the STORE unit encodes the order of the sequence. During external input presentation LSTM-STORE learns a sequence from external teacher in a supervised manner. In absence of external input STORE unit trains the LSTM network for encoded information and the information recalled by STORE to train LSTM could be in veridical as well as non-veridical order.

The Phase Diagram (Fig. 5) shows three key phases for the proposed model. And we will explain these phases in terms of the working of LSTM-STORE model in detail ahead.

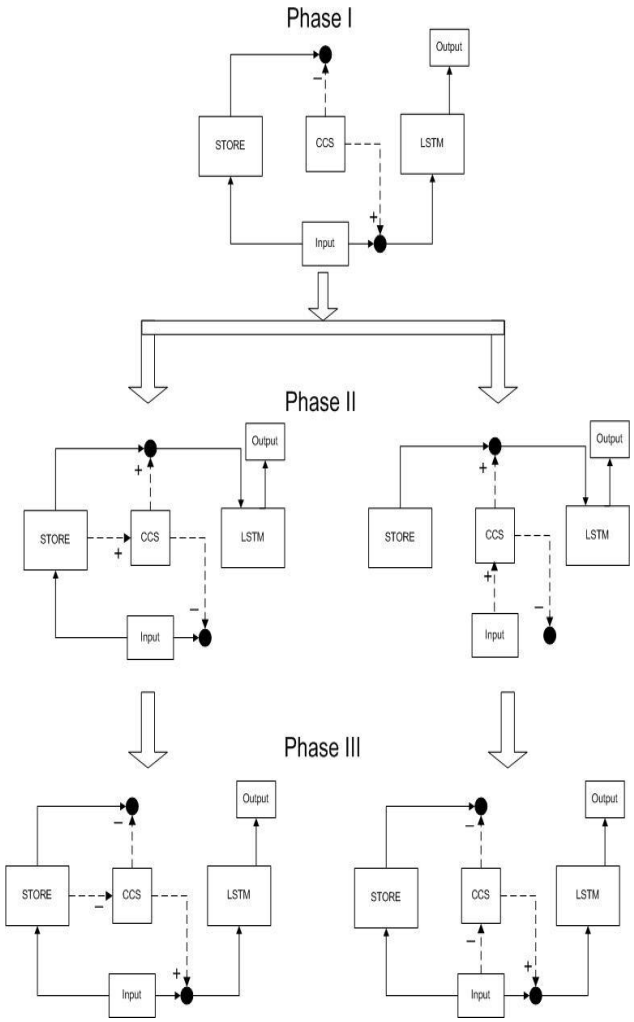


Figure 5: The Phase Diagram, showing the working of the LSTM-STORE model in three main phases

Phase I: Initially the model is in Phase I: the supervised learning mode. LSTM and STORE are both receiving external input, while STORE is recording the temporal order of the input pattern, the LSTM unit is learning to classify the presented input pattern. The Consolidation Control System (CCS) is not receiving any input though it

is sending of signal to inhibit the connection between STORE and LSTM unit, thus STORE will not be able to send any contradictory input signals to LSTM. Similarly it is sending positive signals to allow un-hindered flow of external input to LSTM unit.

Phase II: The LSTM-STORE model will continue to function in the Phase I status unless either one of the two conditions occur: a) either the external input patterns ends and a no new input pattern takes its place or b) the STM activation across all nodes in STORE unit reach their maximum value, in other words STORE exhausts its storage capacity and has to let go of this data. This fact correlates with the natural phenomenon of limited capacity of short term memory in humans.

When either a) or b) condition occurs, CCS receives triggering signals, either from STORE unit or from Input unit. These signals reverse the output signals of CCS unit to the gate units. These gate units (as shown in Fig. 4 and Fig. 5) guard the connections between STORE and LSTM units, and Input and LSTM units. Thus the signals from CCS restrict the flow of external input to LSTM unit, and allow the flow of information from STORE to LSTM unit. Phase II depicts the unsupervised learning phase of proposed system. Previous information now with non-veridical order is used to train the proposed model.

Phase III: To precipitate to the next phase as shown in Fig. 5 the CCS receives again either one of two inputs (either from STORE or Input unit). The signal from Input unit indicates that next input pattern has arrived, and the signal from STORE unit indicates that it has released all its activation and reset its internal state. These negative inputs restore the output signals emitting from CCS to their original state. That is positive signal (releasing the flow) to gate connecting the Input and LSTM unit, and a negative signal to the gate connecting the STORE and LSTM unit. This marks the beginning of Phase I and the cycle continues.

In the proposed solution LSTM-STORE, the internal as well as external triggers have the ability to switch the system to unsupervised learning mode from supervised learning mode. In Phase I the external input is teaching LSTM-STORE model an information pattern in a supervisor manner; trigger from either the input unit (external environment) or STORE unit (internal functioning body) could transform LSTM-STORE model's state to unsupervised learning mode. The role of teacher is taken over by an internal unit. The added advantage is that even in absence of external input our system never stops learning and keeps on perfecting the knowledge acquired.

There are certain assumptions that we are making while proposing this model, these assumptions are probable in consideration with the nature of cognitive tasks. One assumption is that the inter-input duration is mostly less then or equal to intra-input delays i.e.:

$$\alpha \leq \beta \qquad (12)$$

The reason is that the proposed system should have sufficient time to stabilize its learning and although the system could work even if the above situation does not hold its working will not be as efficient.

Another constraint is a finite threshold value S set to curtail the number of items stored in the working memory; the purpose is to conform to the similar biological constraints in humans. The STORE nodes storing the activity pattern are denoted by x1, x2,.., xn. The total STM activity across STORE unit at anytime ti is given as:

$$Si = \sum_{k=1}^{i} x_k(t_i) \qquad (13)$$

An input sequence is presented to the LSTM-STORE model through the time interval [α-ti, ti]. The input sequence is fed simultaneously to both the LSTM and STORE unit, LSTM while learns the sequence through the truncated Back-Propagation Through Time (BPTT) algorithm [20], STORE encodes the temporal order of the sequence.

The affect of I is stored by the activity pattern across the STORE network; this activity pattern at any time tj in given as: x1(tj),x2(ti),… xk(tj)… At this instance the activity pattern across STORE will contain the affect of previously presented activity pattern and will be of the form: x1(ti), x2(ti),… xk(ti)… Now the previous input pattern will be recalled and will be provided to the LSTM unit for rehearsal. The temporal order of this input sequence will now change; and now it will either present primacy order if:

$$xk\text{-}1(ti) > xk(ti) \qquad (14)$$

or it will present recency if:

$$xk(ti) > xk\text{-}1(ti) \qquad (15)$$

or bowing which combines the affect of both primacy and bowing.

According to variance principal which is based on human cognition, for longer lists the recent items are recalled first. Therefore A; which determines order of encoding in STORE model is set to A>1 for recency. As LSTM are designed to cater to lists with longer time lags, the value of A is also modulated by the Control Unit.

The delay between two input sequences is given as β. Therefore during the time interval [ti+1, ti + β] in absence of external input pattern or when the Si >= S, the rehearsal phase begins. At this phase Consolidation Control System (CCS) sends signals to gate units G1 and G2; G1 gets a positive boosting signal, and G2 get a negative inhibiting signal (see Fig. 5). The LSTM unit's input equations 1, 2, 3 given in section III take the form:

$$net_{out}(t) = \sum w_{out-k} x(t-1);\ z^{out}(t) = f_{out}\left(net_{out}(t)\right) \qquad (16)$$

$$net_{in}(t) = \sum w_{in-k} x(t-1);\ z^{in}(t) = f_{in}\left(net_{in}(t)\right) \qquad (17)$$

$$net_c = \sum w_{ck} x(t-1) \quad (18)$$

The actual input order was y1(1), y2(2),…yn(n), now this sequence becomes: yk(ti+1), yk-1(ti+2),… this sequence along with the original target values is fed to the LSTM network, the LSTM weights are optimized with respect to that.

6. Comparative Experimental Results

Task: The task selected to test the above described architecture was simple but sufficient to test the suitability of our proposed solution with respect to lower cognitive reasoning task. These tasks are input storage, recall (veridical as well as non-veridical order) and processing. The LSTM and LSTM-STORE were trained for a given input sequence in specific order. For testing, the same sequence was presented to both the LSTM and LSTM-STORE network out of order to check if the networks could classify the sequence, and how efficient the classification process will be. The input sequences had the long time lags that only LSTM architecture could handle adequately.

Comparison: The training and testing sequences used for both the LSTM and LSTM-STORE architectures were similar, and results were obtained for both the models for same parameters and input target sequences.

Architecture: A simplest possible setup was implemented here to test the model; STORE1 was chosen as working memory to encode order. It has two layered architecture, STORE1 can not only successfully encode the invariance principal [3] but it is also robust as the input durations do not affect the stored activity pattern. This is suitable for the type of input patterns for which LSTM is termed effective. The number of input and output unit was set as 1, and output layer was biased. The number of memory cell blocks for LSTM was set at 2, with each block having size 2; learning rate was set at 0.1 for our experiment.

Results: One important thing to note before moving to the result is that the initial weights for LSTM unit's connection are randomly chosen between the range [-0.1, 0.1]. And for the connections between CCS, LSTM and STORE units the weights alternates between 0 and 1 according to learning phase. Also the training set and testing set both were unique; with each set ten trials were conducted with different initial weights (all weights are between the above mentioned range of [-0.1,0.1]). Similar situations were used to test both the LSTM and LSTM-STORE architectures. Interestingly where LSTM cannot perform well LSTM-STORE models learn to solve the task with improved results.

The results of the ten trials for LSTM and LSTM-STORE architecture are given in the Table 4.1:

Table 1: The table 1 shows the reduced number of epochs in case of LSTM-STORE architecture while in comparison with the original LSTM architecture. Also the LSTM-STORE achieves success in classifying sequences earlier then the LSTM for the test set

Trial No.	Total Epochs to reach the set value of MSE* for LSTM	Success at Epoch (the epoch at which the sequence is successfully classified for LSTM)	Total Epochs to reach the set value of MSE for LSTM-STORE	Success at Epoch (the epoch at which the sequence is successfully classified for LSTM-STORE)
1	101	101	60	49
2	128	127	74	64
3	121	115	68	57
4	111	106	63	52
5	125	116	70	59
6	124	119	70	59
7	109	105	63	52
8	111	108	64	53
9	122	116	68	57
10	115	111	65	54
MSE: Mean Square Error, value set same for LSTM and LSTM-STORE				

.

7. Conclusion

Type your main text in 10-point Times, single-spaced. Do not use double-spacing. All paragraphs should be indented 1 pica (approximately 1/6- or 0.17-inch or 0.422 cm). Be sure your text is fully justified—that is, flush left and flush right. Please do not place any additional blank lines between paragraphs.

This research work is an attempt to model the lower level cognitive tasks. The focus remains on modeling ability of brain to learn new information with the aid of external teachers and to learn derived knowledge in an unsupervised manner based on the existing learned knowledge. The approach of modeling adopted was of neural networks, as neural networks are simulators of the neural behavior of brain due to their neural plausibility [16] they can imitate several cognitive behaviors with ease as compared to symbolic solutions.

By augmenting Long Short Term Memory with a working memory like Sustained Temporal Order Recurrent, which is designed to handle temporal data, we can improve the performance as shown in Fig. 6. The average increase in performance is 57%, if we compare

the results obtained from LSTM and LSTM-STORE architecture.

The implications of this are quite significant both with respect to cognitive modeling of lower level reasoning tasks but also with respect to language processing, hand writing recognition and speech and vision processing which are categorized as higher level cognitive reasoning tasks. We can improve and optimize the neural network architectures being employed to model these tasks by instilling in these networks the underlying processes like perception and storage.

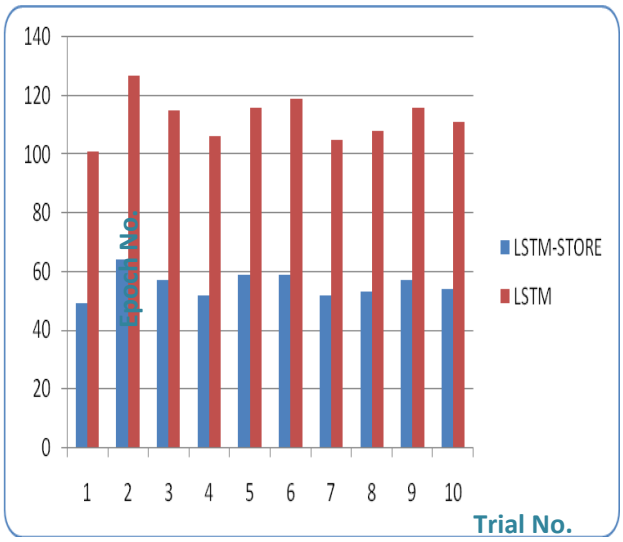


Figure 6: Chart comparing the performance of LSTM and LSTM-STORE architecture with the criteria: at which epoch network learned to classify all input sequences with out any misclassification. At X-axis we have the Trial No. and at Y-axis we have the epoch number.

For future work the architecture could be improved and tested with different type of input data especially real world input tasks like speech, and visual processing. Further experiments could be performed to verify several ideas proposed by cognitive psychology researchers. Certain improvements in the architecture could also be proposed to enable it to suit better to higher level cognitive tasks and compare performance with LSTM model

8. References

[1] Alac, M., "Squashing, Rotating, Seeing, and Going: On Visual Knowledge in fMRI Research", *Proc. of the 25th Annual Conference of the Cognitive Science Society Part 2*. Boston, Massachusetts. pp 73-78, 2003.

[2] Atkinson, R., and Shiffrin, R., "Human memory: A proposed system and its control processes", *K Spence & J Spence (Eds.). The psychology of learning and motivation: Advances in research and theory (Vol. 2)*. New York: Academic Press 1968.

[3] Bradski, G., Carpenter G. A., Grossberg S., "STORE working memory networks for storage and recall of arbitrary temporal sequences", *Biological Cybernetics*, 71(6), October, pp 469-480, 1994.

[4] Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J. H. & Rosen, D. B., "Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps", *IEEE Transactions on Neural Networks*, 3, pp 698-713, 1992.

[5] Craik, F. & Lockhart, R., "Levels of processing: A framework for memory research". *Journal of Verbal Thinking and Verbal Behavior*, 11, 671-684, 1972
.

[6] Gers F. A., Schmidhuber J., "LSTM Recurrent Networks Learn Simple Context Free and Context Sensitive Languages", *IEEE Transactions on Neural Networks* 12(6) 1333-1340, 2001.

[7] Graves A., Liwicki M., Fernandez S., Bertolami R., Bunke H., Schmidhuber J., "A Novel Connectionist System for Improved Unconstrained Handwriting Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, pg. no. 5, 2009.

[8] Graves A., Schmidhuber J., "Frame-wise phoneme classification with bidirectional LSTM and other neural network architectures", *Neural Networks* 18:5-6, pp. 602-610, 2005.

[9] Grossberg, S., "A Theory of Human Memory: Self-organization and Performance of Sensory-motor Codes, Maps, and Plans," *Rosen R, Snell F (Eds) Progress in theoretical biology*, Vol 5. Academic Press, New York, pp 233-374, 1978.

[10] Hochreiter S. & Schmidhuber. J., "Long Short-Term Memory", *Neural Computation*, 9(8):1735-1780, 1997.

[11] Klapper-Rybicka M., Schraudolph N. N., Schmidhuber J., "Unsupervised Learning in LSTM Recurrent Neural Networks", *Proc. International Conference on Artificial Neural Networks*, pg. 684-691, 2001.

[12] McClelland, J. & Rumelhart, D., "Parallel distributed processing", *Explorations in the microstructure of cognition* Vol. 2. Psychological and biological models. Cambridge, MA: MIT Press, 1986.

[13] Pachalska, M. Weber, M., "Neuropsychology and philosophy of mind in process: Essays in honor of Jason W. Brown", *ontos verlag*, pp 44-48, 2008

[14] Rumelhart, D., "Schemata: The building blocks of cognition", *R. Spiro, B. Bruce, & W. Brewer (Eds.)*, Theoretical issues in reading comprehension. Hillsdale, NJ: Erlbaum 1980.

[15] Silva S. and Caetano., "Using Artificial Recurrent Neural Nets to Identify Spectral and Spatial Patterns for Satellite Imagery Classification of Urban Areas" Kanellopoulos I, Wilkinson GG, Roli F, Austin J (eds), *Neurocomputation in Remote Sensing Data Analysis*, Springer-Verlag 151-159, 1996.

[16] Stillings, N. A., et. al.. "Cognitive Science: An Introduction". MIT Press, 1995.

[17] Vieira, F. H. T. and Lee, L. L., "A Neural Architecture Based on the Adaptive Resonant Theory and Recurrent Neural Networks", *Int. Journal of Computer Science and Applications*, Vol. 4 Issue 3, pp 45-56, 2007.

[18] Wang, D. L. Yuwono, B., "Incremental learning of complex temporal patterns". *IEEE Transactions on Neural Networks*, 1465-1481, 1996.

[19] Werbose P. J., "Generalization of Backpropagation with application to a recurrent gas market model", *Neural Networks*, 1, 1988.

[20] Williams R. J. and Zipser, D., "Gradient based learning algorithms for recurrent networks and their computational complexity", *Backpropagation: Theory, Architectures and Applications*, Hillsdale, NJ: Erlbaum 1992.

Applying prediction methods for nonstationary time series from an Distributing Company

Jeuel Bernardes Alves – jeuel@pucgoias.edu.br
Pontifical Catholic University of Goias

Celso G Camilo-Junior - celso@inf.ufg.br
Institute of Informatics,
Federal University of Goias

Abstract

This work addresses the challenge of supplying the stock of an organization based on peculiar behavior of each unit sale. For that, sales data from a single product of an Distributing Company, whose performance is derived by the efforts of seven subsidiary units, are used for performance analysis of some methods of prediction in this context. The Poisson method was used to calculate the probability of meeting all demand for the stock desired. The Moving Averages (MM), Holt-Winters (HW) and Elman Neural Network (ANNe) were applied as prediction methods. Based on the mean square error performance the methods were compared. As a result, the HW method performed best, followed by MM and ANNe.

KeyWord: Elman Neural Network, Holt-Winters, Moving Average, Forecasting

1 Introduction

Data Mining (DM) is an area of computing that lets you work in the discovery of new information based on patterns or rules with a large volume of data (Navathe, Elmasri, 2005). The terms Data Mining and Knowledge Discovery in Databases - KDD are commonly mistaken as synonymous. However, DM is a part of the KDD process (Calil et al., 1996) (zuchini, 2003).

Among the various targets of DM and KDD, the prediction analyzes how certain variable will behave, by other related attributes (Fayyad et al., 1996). The forecast is not an end in itself but a means of providing information and support for a consistent decision making, to achieve certain goals (Morettin, 1981).

One application is the prediction of time series forecasting, which is to provide the series value at time t + h from a predictive model built based on a set of information from a series collected up to time t, Equation (1) .

The predictions of a time series can be classified as short, medium or long term, depending on the forecast horizon. Thus, two different techniques to predict future values of a time series can be used (Mueller, 1996).

- Multi-step forecast: where the set of current values is used to predict a given instant, the value of the next moment is determined from the current values plus the predictions already made . This technique is adopted for long forecast horizons, as it seeks to identify trends and turning points in the most relevant series;
- Forecast simple step: where the prediction is made only for the time period immediately following the current, from observations of the current series.

According to Souza (1989), ensuring the quality of forecasts of a time series is only achieved by adopting the forecast horizon as the instant of time immediately subsequent to the origin t.

A variety of methods of DM for pattern recognition, and some of these are of Artificial Intelligence. Among those, this paper uses the technique of Artificial Neural Network (ANN) as a tool to create a prediction model (prediction) of a series. We opted for the ANN by good performance elsewhere. Comparisons with other methods of ANNs can be obtained by works (Viglione, 2010), (Maciel; Ballini, 2010).

The other two methods, Moving Average and Holt-Winters, are known and used in problems arising from financial market series, where the required degree of effectiveness is high.

A good forecast of demand causes inventories are well controlled, because from the moment the company become aware of the demand, it can prepare to make their decision-making (Almeida; Passari, 2006), (Flores; Werner, 2007).

Considering the importance of the prediction of sales in decision making, this paper discusses some methods to forecast demand in wholesale distributor in a car parts. Of the various products available for sale, this work deals with the forecast of unit sales of the product 6167, considered one of the most important financially for the company.

This paper is organized as follows. In section 2, we summarized the concept of time series. In section 3 the forecasting methods used in this work are presented. In addition, a presentation of the problem is done in Section 4, testing and analysis of the results are presented in Section 5 and finally, the conclusion is described in section 6.

2 Time Series

With examples of definitions, refer to: "A set of observations taken at certain times, usually at equal intervals" (Spiegel, 1993). "It consists of a sequence of values measured at equal intervals of time " (Han and Kamber, 2001).

A time series analysis is intended to study

the relationship of temporal data in the same record, the volatility of these over the period observed and the construction of models that represent them in an attempt to develop predictions for the following periods.

According to Ehlers (2009), many of the properties observed in a time series Xt can be captured by assuming the following decomposition

$$X_t = T_t + C_t + R_t \quad (01)$$

where T is a trend component, Ct is a seasonal and cyclical component or Rt is a random or noise component (the part not explained), ie.

$$\cdots = C_{t-2s} = C_{t-s} = C_t = C_{t+s} = C_{t+2s} = \ldots . \quad (02)$$

Thus, periodic variations can be captured by this component. In general, one way to deal with data containing a seasonal trend is to fit a polynomial equation,

$$X_t = \beta_0 + \beta_{1t} + \cdots + \beta_{p\,tp} + \varepsilon_t \quad (03)$$

To Migon (2010), the main objectives of a time series analysis are:

- Understanding the mechanism generating the series;and
- Predicting the future behavior of the series.

From these objectives, it is possible to mitigate the forecast errors, with application of forecasting methods based on simplified models of the problem in order to promote the enrichment of meaningful alternatives in the process of strategic decision.

3 Forecasting methods

3.1 Moving Average

In the moving average model becomes an arithmetic average of n most recent data for the series. Each new observation of a new real data becomes a new medium:

$$X_{n+1} = \frac{(X_1 + X_{2+\ldots+} X_n)}{n};\ i,n > 0 \quad (04)$$

The predictions are defined from a range of historical data of n input-shift structure in the sequence of historical values. Its input parameters for performance tuning of the structure is the breadth of data and the weights you enter the projection.

Among the works that use moving averages as forecast cites are:

- Gujarati (2006) indicates the econometric models that use moving averages to predict, as the models ARIMA (Auto-regressive Integrated Moving Average), ARMA (Auto-regressive Moving Average) and NARMA (Non-linear Auto-regressive Moving Average) ;
- In works such as Villanueva (2006) are applied in studies of machine processes using time series prediction competition models with moving averages;
- Feliciano (2009), adapting the work of Schroeder (1993), considers moving averages in their framework of options for methods of predicting demand with integrated distribution system;
- Dutra (2009, et al.) in your article uses moving averages to determine the index of seasonality to verify the behavior of the series of demand for biodiesel.

3.2 Holt-Winters

The method of Holt-Winters (HW) is based on three equations, one for each component: level, trend and seasonality (see Equation 05-08).

$$P_t = a_t + b_t + c_t \quad (05)$$

$$a_t = \alpha(X_t - c_{t-m}) + (1-\alpha)(a_{t-1} + b_{t-1}) \quad (06)$$

$$b_t = \beta(a_t - a_{t-m}) + (1-\beta) b_{t-1} \quad (07)$$

$$c_t = \lambda(X_t - a_t) + (1-\lambda) c_{t-m} \quad (08)$$

Where:

a_t: average level series; b_t: parameters of the seasonal part of the model; c_t: random component unexplained by the model;

α, β, λ: exponential parameters straighteners, level, trend and seasonality;

X_t: observed value

Additive model is one where the term representing the periodic variation of the time series has statistical behavior that regardless of growth rate (positive or negative) of the series.

Among the works that use HoltWinter as forecast, it cites:

- Spencer (2005) in their study of energy demand in the electricity sector in the state of Santa Catarina, concludes that apply neural networks ANNs can be replaced by a Holt-Winters model, in that situation.
- Menezes (2010, et al.) develops a different prediction of the tank water supply Leite for the large region of the state capital of Goias until 2025, so using the Holt-Winters method.
- Lucio (2010, et al.) develops a study on the rainfall over the Brazilian territory was used for both the ARIMA and Holt-Winters to extrapolate the behavior of precipitation series with few oscillations.
- Knupp (2007) seeks the establishment of important correlations between the presence of pollutants in the river regions and the sources of contamination were fully completed. This study has also used the Holt-Winters model that made predictions based on the job.

3.3 Elman Neural Networks

Neural networks provide, usually, non-parametric quantitative knowledge. They are suitable for identifying and learning systems that exhibit complex behavior.

Elman networks have been created by Jeffrey L. Elman in 1990 at the University of California. These networks use an architecture of

recurrent (feedback), and the differential is an auxiliary layer called the context. Each context stores the last saved state of the network. The fact of maintaining a historical context layer causes this type of network has good results in prediction of time series.

Elman introduced to ANNs memory (Elman, 1990) to the context units. The input and output units interact with the external environment, while the intermediate units and context do not. In this case the context units are used only to memorize the previous activations of the intermediate units and can be considered as time delay in one step.

The inputs of the network are: ui (k), yj (k), xi (k) and the outputs of layer, in Equations 09, 10 and 11 are presented their respective frameworks.

$$x_j(k) = f\left(\sum_{i=1}^{m} w2_{ij} u_i(k) + \sum_{i=1}^{r} w1_{ij} c_i(k)\right), \quad (09)$$

$$x_i(k) = x_i(k-1), \quad (10)$$

$$y_j(k) = g\left(\sum_{i=1}^{r} w3_{ij} x_i(k)\right) \quad (11)$$

where f (.) and g (.) are functions with outputs linear or nonlinear hidden layer and output layer, respectively.

The Elman Neural Network is generally considered a special type of feedforward ANN with additional memory neurons with local feedback. The feedback connections from hidden layer to context layer makes the input ANNe be sensitive to historical entries which is important in modeling dynamic systems, see Figure 1.

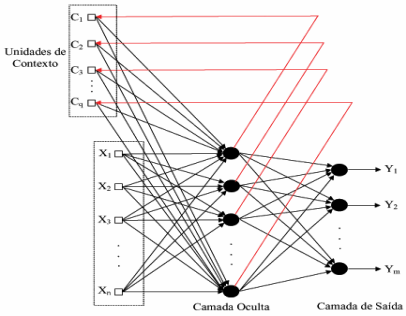


Figure 1: An example of Elman Neural Network architecture illustrated by Segatto

Among the works that use of ANNe as forecasting cites are:

- Segatto (2008) in their work to address the problem of reconstruction of the distorted signals caused by saturation of current transformers is implemented Elman recurrent ANNs used to reconstruct the distorted signals by the saturation of current transformers.
- In the study of Paschoalino (2007, et al.) proposes the use of computational intelligence technique known as Elman Neural Network to forecast electricity consumption in Brazil.
- Carvalho (2007, et al.) in your paper the neural networks are used to model time series of flows of hydroelectric dams, the variables obtained from drainage, steps over the years. Due to the amount of data the use of ANNe was considered a very natural choice because of their ability to predict system performance monitoring of the dam.

4 The Problem

This paper proposes the use of predictive methods to identify the future demands of an organization for a given product. With the demand information, you can set inventory levels required of rouding.

However, before making the prediction is necessary to study the problem in order to understand the behavior of sales, therefore, support the application of the methods.

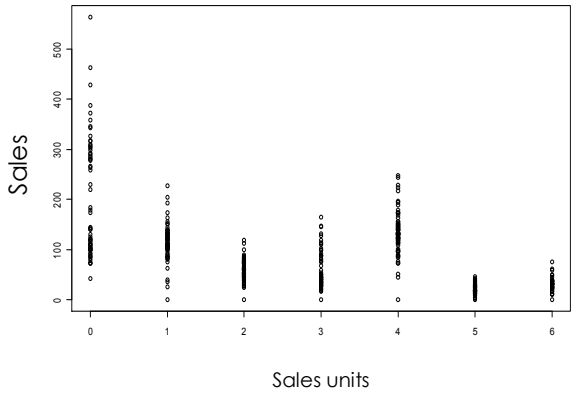


Figure 2: Alignment of all series of sales per sales unit

Table 1: Descriptive analysis of scattering of the series

Units	Min	Max	1stQ.	3rdQ.	Mean	Median	sd
0	42	564	102	291.5	198.65	141.5	116.13
1	25	227	100.5	131	114.69	113	35.04
2	19	118	47	74.25	60.62	60.5	20.11
3	16	164	33	82.5	59.65	44.5	35.81
4	45	247	112.8	151.2	136.18	131.5	41.75
5	3	46	15	28	21.77	19	9.52
6	10	75	24	37	34.13	32	13.49

Table 01 describes the control limits (upper and lower) the performance of companies 1, 2 and 4 have higher centrality of your average sale, this is suggested by evidence of concentration of the medians with the means. Among these, the company introduced two smaller range of control limits. Moreover, units 0 and 3 demonstrate that their average sales are closer to the first quartile, however do not have enough evidence to gauge opinions of behavior without an analysis of the series. Deviations

(sd) of the series of units. Units 2, 5 and 6 record the smallest deviation from the pack. However the units 1,2 and 4 are related to minor deviations from the mean (0.3055, 0.3318 and 0.3066), this means that the diversion of such unit representing 30.55% and 33.18% 30.66%, respectively, about their respective averages.

Following the series of each unit of sales were to test for stationarity, which allows you to check the regularity of the deviations around the average number of sales. The KPSS test is used to check stationarity of the series. Results are presented in Table 02.

Table 2: Performance ratio of stationarity of each unit sale

Units	p-value Kpss	stationarity
0	0.01	no
1	0.02429	no
2	0.01	no
3	0.01	no
4	0.09544	yes
5	0.08174	yes
6	0.01	no

This review process allows a dynamic understanding of the facts presented in Tables 01 and 02, because the series is observed along its trajectory, so it can be seen in the results is evidence of nonstationary behavior observed series under p-value of 0.05. Previously said, the sales units 1,2 and 4 showed signs of predictability, however this test with only 4 unit actually presented evidence of regularity of deviations around the mean.

The method of Poisson series in each sales unit was directed to identify the probability of lost sales due to lack of stock item, considering that a deviation of 0.25 to define the control limits (upper and lower).

Table 3: Performance ratio of lost sales of each unit sale

Units	Probability
0	0.143661
1	0.050536
2	0.032295
3	0.068114
4	0.079079
5	0.023961
6	0.050536

In Table 03 we can observe that unit sales with better results is the unit with 5 and 4 is the worst performance for the application of Poisson considering the parameters set for observation.

Anyway, these steps descriptive analysis can verify signs of a steady problem not composed 74.85% of sales volume with a standard deviation around the mean irregular, and the scenario of events often a lack of goods and services for demand, as indicated by the result ofof applying Poisson . The results of the descriptive analysis shows signs of difficulty in the process of predicting stock in the sale of units rouding.

5 Tests and Results

Based on the number of variables and size of the series it was decided that the scale reading of historical data to infer the following week for three weeks. This amplitude was defined by the cycle of fiscal duties and experience team of sales of the units reported in an interview.

The application procedure considered the same criteria for reading and test methods.The amplitude Xt-n represents the segment of data used to predict the subsequent periods (Xt +1) and its displacement is given by (Xt + k) where k is the size of the shift made to start the next prediction. Since training and testing were respectively the horizon of learning the behavior of the series and finally, the methods set, proceed with testing. At this stage, in the case of ANNe, although adjustments are made for learning achieved during training, since the other methods that fit the learning procedure is not performed because of his mathematical conception. The Holt-Winters method seeks to build a polynomial model of adherence to the training series, the same number of training available to the other methods, and from that model's behavior is extracted from this series that is designed its tests, according to Table 04.

Table 4: Table of comparison of the conditions of training and testing to the methods used

Methods	Xt-n	k	training	Test
MM	3	1	-	20
HW	-	-	48	20
ANNe	3	1	48	20

However, the observation horizon Holt-Winters method was higher, which could penalize them the perception in the short term, given the characteristics of non-stationary series. Another case is the lack of training of moving averages, but given its structure that has no positive effect in a non stationary.

Through analysis of the mean square error of prediction (SSE) obtained from the application of the methods proposed to have a comparison between the methods moving averages, Holt-Winters and Elman Neural Network to the problem addressed. In Table 05 presents the hese the predictions of each method in the series of sales of each unit sale of rounding.

Table 5: Table of performance indicators from the units of squared erros

Methods	Mean	Unit 0	Unit 1	Unit 2	Unit 3	Unit 4	Unit 5	Unit 6
ANNe	15,79	1,48	15,73	16,51	3,37	37,27	29,96	19,54
MM	2,56	0,43	2,21	2,54	3,82	1,9	1,86	5,13
HW	0,91	0,13	0,84	0,86	0,23	0,89	2,13	1,31

As overall average of Table 5, the HolterWinters method was superior to the others, followed by the Moving Average (MM) with SSE 181.32% and ANNe with SSE 1631.87%. So on that conditions, the Holt-Winters method showed better performance than the other methods presented in this work.

6 Conclusion

The period encompassing the beginning and end of the process of production and distribution logistics of the commodity has a term that prevents the availability of the goods within a timely manner.

In order to provide the availability of merchandise, that the prompt delivery, inventory of goods is established. Thus the need to forecast the stock level of merchandise under the expected demand. His dilemma is the availability of merchandise and amount of financial liquidity used in stock, which results in loss of profitability and contribution margin per unit sale.

Therefore the process of prediction is an inevitable procedure for units producing or providing services associated with service demand. In the case of wholesale unit, still aggravating to differentiate itself by prompt delivery, need to supply their stores, whose goods are produced in different units scattered suppliers in Brazil and the world with production schedules and delivery are also different, so forecasting is essential to reduce delays and facilitate the availability of commodity demand.

This study considered methods widely used in solving problems of time series prediction (MM, HW and ANNe), that sought to identify the most performing stock prediction of the Distributing Company. Given the equivalences, the results demonstrated the significant advantage of the Holt-Winters method in the process of predicting the sales of an organization's product review.

For future works, we suggested to broaden the scope of the problem to an analysis of more products or product groups, with the intention of working with multivariate analysis methods. Moreover, we suggested using other ANN algorithms, e.g. static learning rate with Backpropagation [21].

7 References

[1] ANGELO,Claudio Felisoni de, OLIVEIRA,Mauri Aparecido de, FÁVERO, Luiz Paulo Lopes. Aplicação de Métodos de Ajustamento Sazonal em Séries Temporais - VI SEMEAD, FEA-USP, site: w*ww.ead.fea.usp.br , acessado 21/03/2011;*

[2] DUTRA, Nadja Glheuca da Silva; BARROS NETO,Júlio Francisco, ALBERTIN, Marcos Ronaldo, PONTES, Heráclito Lopes Jaguaribe, DO CARMO,Breno Barros Telles. Biodiesel forecast evaluation using a forecasting model based in diesel demand. Associação Brasileira de Engenharia de Produção – ABEPRO, Revista de Produção ISSN 1676 - 1901 / Vol. IX/ Num.III/ 2009 da Universidade Federal de Santa Catarina – UFSC http://www.producaoonline.org.br, acessado em 24/03/2011 as 17:35h;

[3] EHLERS, RICARDO S., Análise de séries temporais, http://www.icmc.usp.br/~ehlers, 06/03/2010 18:24h – 5ª.edição, São Paulo -2010;

[4] FELICIANO, Ricardo Alexandre. Uma proposta de gerenciamento integrado da demanda e distribuição, utilizando sistema de apoio a decisão (SAD) com business intelligence (BI) http://www.teses.usp.br , acessado em 24/03/2011 as 17:03h;

[5] GUJARATI, Damodar N., Econometria básica / Damodar Gujarati;tradução de Maria José Cyhlar Monteiro. – Rio de Janeiro: Elsevier, 2006 – 4ª.reimpressão;

[6] KNUPP,Eliana Aparecida Nonato. Uso de métodos estatísticos para dados de qualidade de águas: estudo de caso, Rio das Velhas. Tese de Doutorado em Ciências, 2007 – Química, UFMG – ICEx/DQ. 651a T. 254a, http://biblioteca.cdtn.br, acessado em 24/03/2011 as 19:02h;

[7] LUCIO, Paulo Sérgio, SILVA, Fabrício Daniel Dos Santos, FORTES, Lauro Tadeu Guimarães, SANTOS, Luiz André Rodrigues dos, FERREIRA, Danielle Barros, SALVADOR, Mozar de Araújo, BALBINO, Helena Turon, SARMANHO, Gabriel Fonseca, SANTOS, Larissa Sayuri Futino Castro dos, LUCAS, Edmundo Wallace Moteiro, BARBOSA, Tatiane Felinto, DIAS, Pedro Leite da Silva. A combined stochastic model for seasonal prediction of precipitation in Brazil - Revista Brasileira de Meteorologia, v.25, n.1, 70 - 87, 2010, http://www.scielo.br, acessado em 24/03/2011 as 18:03h;

[8] M. Garofalakis, R. Rastogi e K. Shim, “Mining Sequential Patterns with Regular Expression Constrain”, IEEE Transaction on Knowledge and Data Engineering, Vol. 14, No.3, Ma/June 2002;

[9] MADDALA, E., Econometrics, McGraw-Hill, Nova York, 1977;

[10] MANSSUR, Luiz Henrique. Aplicação de um modelo de previsão de demanda em uma empresa de agrobusiness. 2006 - www.pro.poli.usp.br*, acessado na data 23/03/2011;*

[11] MATSURA, E. (2007). *Comprar ou vender?Como investir na bolsa utilizando análise gráfica* (5a ed.). São Paulo: Saraiva;

[12] MENEZES, Jose Elmo de, BARROS,Victor Freitas de Azeredo. A utilização do método de Holt-Winters na estimação do consumo de água na grande Goiânia para as próximas décadas - IV Seminário de Iniciação Científica, IV Seminário de Iniciação Científica do Instituto Federal de Educação, Ciência e Tecnologia de Goiás/IFG – 2010, http://revistas.ifg.edu.br, acessado em 24/03/2011 as 18:03h;

[13] MIGON, Hélio, Análise de séries temporais, UFRJ/Instituto de Matemática/Métodos quantitativos - site http://acd.ufrj.br, 06/03/2010 16:56h;

[14] MORETTIN, Pedro A. Econometria financeira Um curso em séries temporais financeiras , Departamento de Estatística/IME, São Paulo 2004;

[15] MURRAY R. Spiegel, Estatística, Terceira Edição, Makron Books, 1993;

[16] R. Agrawai e R. Srikant, "Mining Sequential Patterns", Proc. 11th Int'l Conf. Data Eng., March 1995;

[17] SEGATTO, Ênio Carlos,COURY, Denis Vinicius Redes neurais aplicadas a relés diferenciais para transformadores de potência - Revista Controle & Automação/Vol.19 no.1/Janeiro, Fevereiro e Março 2008;

[18] VILLANUEVA, Wilfredo Jaime Puma. Comitê de Máquinas em Predição de Séries Temporais – trabalho apresentado em 2006, ftp://ftp.dca.fee.unicamp.br, acessado em 24/03/2011 as 16:30h;

[19] CARVALHO, Janaina Veiga, ROMANE, Celso. Redes neurais temporais aplicadas ao monitoramento de barragens. RESI-Revista eletrionica de Sistemas de Informaçnao Edição 10, No1 2007, http://revistas.facecla.com.br, acessado em 25/03/2011 as 00:45h;

[20] PASCHOALINO, Fernanda Fidelis, LOUREIRO,Tabita Yaling Cheng, MELLO, João Carlos Correia Baptista Soares de, NETO,Luiz Biondi. Previsão de demanda de energia elétrica no Brasil utilizando redes neurais de Elman - SPOLM 2007 ISSN 1806-3632 Rio de Janeiro, Brasil, 8 e 9 de novembro de 2007, http://www.uff.br, acessado em 24/03/2011 as 22:45h.

[21] CAMILO, Celso; YAMANAKA, Keiji. A Practical Method for Finding an Efficient Static Learning Rate for ANN. Proceedings of the the 2008 International Conference on Artificial Intelligence, ICAI 2008, July 14-17, 2008.

Forecasting Generation Waste Using Artificial Neural Networks

Elmira Shamshiry[1], Behzad Nadi[*2], Mazlin Bin Mokhtar[1], Ibrahim Komoo[2], Halimaton Saadiah Hashim[1], Nadzri YAhya[3]

(1)Institute for Environment and Development (LESTARI), Universiti Kebangsaan Malaysia (UKM)

(2) Southeast Asia Disaster Prevention Research Institute (SEADPRI) University Kebangsaan Malaysia (UKM)

(3) Director General, National Solid Waste Management Department, Ministry Of Housing and Local Government, Malaysia

* nadibehzad@gmail.com

Abstract. Municipal solid waste (MSW) is the natural result of human activities. MSW generation modeling is major significant in municipal solid waste management system planning. Predicting the amount of generated waste is difficult task because it is affect by various parameters. In this research, Artificial Neural Network (ANN) was trained and tested to weekly waste generation (WWG) model in Sari's city of Iran. Input data is consisting WWG observation and the number of trucks, personnel and fuel cost were obtained from Sari Recycling and Material Conversion Organization. The gathering data related to monitoring 2006 to2008.

Key Words: Weekly Waste Generation, Prediction, Artificial Neural Network, Sensitive Analysis.

1 Introduction

One of the after-effects of human activities is the Solid Waste (SW). A suitable management system should be developed so that the environmental pollution won't endanger people's health. It's an uphill struggle to implement such a system because of the complicated and wide-ranging nature of the waste. Working out the amount of waste produced is of vital importance to set up the Solid Waste Management System. (SWMS) Being aware of the quantity produced can work wonders due to estimating the amount of investigation in the field of machinery, onsite storage containers, transition stations, disposal capacity and proper organization. There are many different ways to assess the waste generation (WG) rates; the most instrumental of them are load-count analysis, weight-volume analysis and materials-balance analysis. Nevertheless, these are the central strategies for estimating the figures of generated waste, but there are some disadvantages. For instance, the load-count analysis method pinpoints the rate collection, but not explaining the rate of production. The materials balance analysis method is also in the throes of many errors if the source of WG be a massive size (such as a city). On the other hand, traditional methods for calculating the amount of produced solid waste are established, mostly, on the basis of some elements such as population and social-economic parameters of a society and they are computed according to the generation coefficient per person.

These methods mostly comprise some models, classic statistical methods and many new techniques like time series methods and artificial neural networks. In this study, an Artificial Neural Network (ANN) was trained and tested to weekly waste generation model (WWG) for Sari city which is the capital of Mazandaran province in Iran.

The ANN models are basically based on the perceived work of the human brain. The artificial model of the brain is known as ANN (Sahin, et al., 2005). For this reason, ANNs have been usefully applied to a wide variety of problems that are convoluted, define, and gauge; for example, in finance, medicine, engineering, etc. Recently, ANNs have been used in the management of MSW such as a proposed model based on ANN to foresee the rate of leachate flow rate in places of solid waste disposal in Istanbul, Turkey (Karaca & Ozkaya, 2006), the prediction of energy content of a Taiwan MSW using multilayer perception neural networks (Shu et al., 2006), HCl emission characteristics and back propagation prediction by neural networks in MSW/coal co-fired fluidized beds (Chi et al., 2005), recycling strategy and a recyclability assessment model based on an ANN (Liu et al., 2002) and the

prediction of heat production from urban solid waste by ANN and multivariable linear regression in the city of Nanjing, China (Dong, et al., 2003), have been evident in current practice.

Also in the other environmental problems like air pollution (Sahin, et al., 2005 ; Lu, et al., 2004 ; Lu, et al., 2006), surface water pollution (Sahoo, et al., 2006 ; Shrestha & Kazama, 2007), the ANNs have been used. The result of all this research has presented the high performance of ANN in prediction of various environmental factors.

2 Data and Materials

Estimating and forecasting the quantity of producing solid waste depends on the different factors such as geographical situation, seasons, collection frequency, onsite processes, and food habits of the people, economic conditions, recovery and reuse boundaries, existing laws and the cultural conditions of the people. And these factors aren't accurate measurement, and it cannot be used as a standard precise analysis.

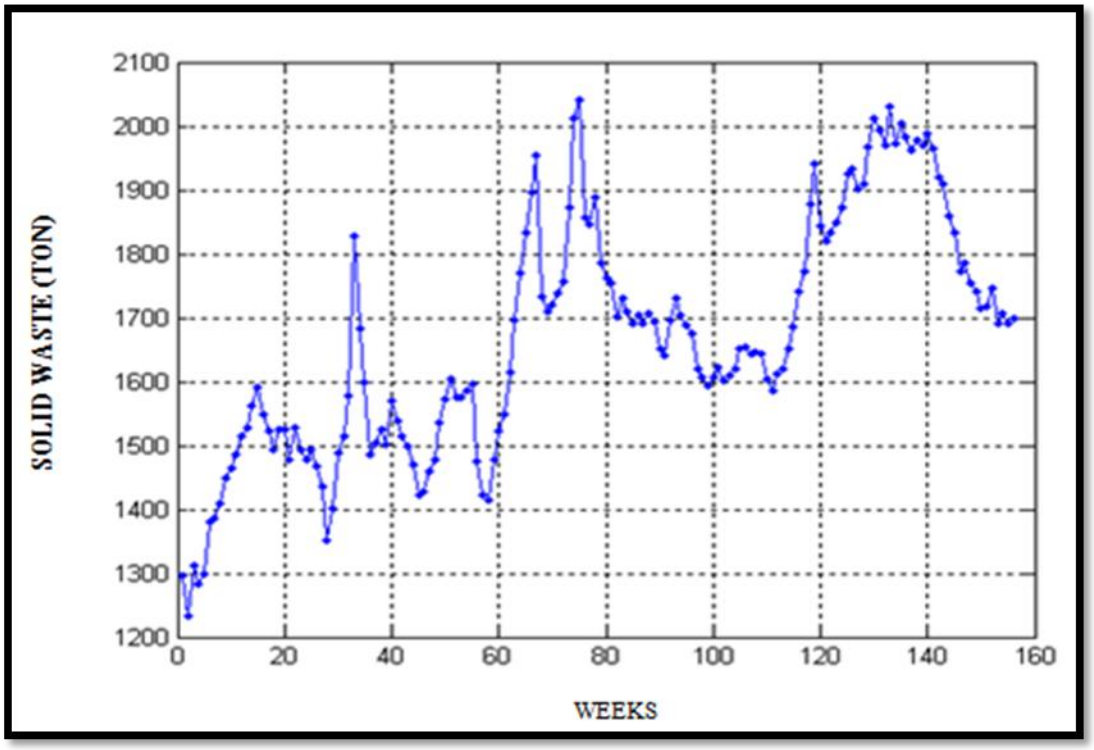


Figure 1. Weekly fluctuation of waste generation in Sari (2006-2008)

The most method to achieve the prediction exact amount of production of solid waste is mostly emphasis on the parameters such as; weekly produced average solid waste, weekly fuel use for collection solid waste, weekly truck use for collection solid waste and number of personal collect solid waste in the Average weekly,

3 Methodology

In this research, Artificial Neural Network (ANN) was trained and tested to weekly waste generation (WWG) model in Sari's city of Iran. Input data; consist of WWG observation and the number of trucks which carry waste, number of personnel and fuel cost were obtained from Sari Recycling and Material Conversion Organization. The

monitoring data from 2006 to2008 are designed to provide the requirements of training and testing the neural network.

In this model, the weight of waste in t+1 week (Wt+1), is a function of waste quantity in t (Wt), t-1 (Wt-1)… t-11 (Wt-11) weeks. Another input data, consist of the number of truck, which carry waste in the week of t (Trt), number of personal (Pr), cost and fuel (Fu). In this research used tangent Hyperbolic function for output layers and to input layers used function Y=X.

In this study, a neural network is trained and tested through the use of software Neural Network version 2.5 and Matlab version 7.8. In this method, the data are divided into 3 parts. The first part of the data is related to network training, the second part is used for stopping calculations when the integrity error starts to increase and the third part is used for the network integrity.

To examine the performance of the ANN model, four statistical indexes are used: the Mean Absolute Error (MAE), the Mean Absolute Relative Error (MARE), the Root Mean Square Error (RMSE) and correlation coefficient (R2) values that are derived in statistical calculations of observation in the model output predictions.

4 Results & Discussion

The statistical analysis of waste materials in Sari during the different seasons between "2006-2008", is presented in Table 0-1. Because the average and median amount is so close to each other, the waste generation in Sari shows a normal distribution through ought the different seasons. The large size of Standard deviation reveals a large generation fluctuation in different seasons of the year during the period of the study.

This great impact could be explained by hobbies and entertainments people have at certain periods and the economic conditions. Different structures of feed forward ANN with three layers and a different number of neurons in the hidden layer were investigated to achieve the best ANN structure for estimating generated waste. Estimating and forecasting the quantity of producing solid waste depends on to the different factors such as cultural, social, living, etc. And these factors aren't accurate measurement and it cannot be used as a standard precise analysis.

At last with respect to MAE, MARE, RMSE and R2 suitable models were selected for the study. Network structure includes the 15-input, 1 hidden layer, and 1-output. In order to estimate the number of neuron in hidden layer for the best prediction, hidden layers from 1 to 10 Neurons were experimented. Evaluation results of ANN with different numbers of neurons in hidden layer are as following Table 2.

Table 2 . Calculated Errors for ANNs with different neurons in hidden Layers applied in training and testing data sets

	Training set				Testing Set			
NO	**MAE**	**MARE%**	**RSME**	**R2**	**MAE**	**MARE%**	**RSME**	**R2**
15-10-1	*90.58*	*0.049*	*114.4*	*0.954*	*121.0*	*0.19*	*159.3*	*0.905*
15-9-1	*48.69*	*0.026*	*69.2*	*0.982*	*72.9*	*0.11*	*106.5*	*0.962*
15-8-1	*35.63*	*0.019*	*54.9*	*0.989*	*55.2*	*0.09*	*84.4*	*0.975*
15-7-1	*40.56*	*0.021*	*55.8*	*0.988*	*46.9*	*0.07*	*66.7*	*0.984*
15-6-1	*38.56*	*0.021*	*56.1*	*0.988*	*45.9*	*0.07*	*65.6*	*0.985*
15-5-1	*40.91*	*0.021*	*51.8*	*0.990*	*57.7*	*0.09*	*77.5*	*0.980*
15-4-1	*26.75*	*0.014*	*44.3*	*0.993*	*48.9*	*0.07*	*68.1*	*0.983*
15-3-1	*34.24*	*0.018*	*46.8*	*0.992*	*46.6*	*0.07*	*71.5*	*0.980*
15-2-1	*37.04*	*0.020*	*55.2*	*0.989*	*38.5*	*0.06*	*51.7*	*0.989*
15-1-1	*33.74*	*0.018*	*45.7*	*0.992*	*60.3*	*0.10*	*81.4*	*0.975*

Summation of errors for each network presented that after increasing number of the neurons in hidden layers, the error increased. However, all tested ANN structures have presented high accuracy. According to the results ANN structure with 2 neurons in hidden layers was selected as an optimal one which can minimize the calculation. However, ANN with 3 to 6 neurons in hidden layers can be employed as the optimal system.

In the first step, a neural network with one neuron in hidden layer was applied. Training and testing data sets are applied to the ANN. The correlation coefficient for training and testing is 0.992 and 0.975, respectively. Presented results indicate high accuracy in prediction.

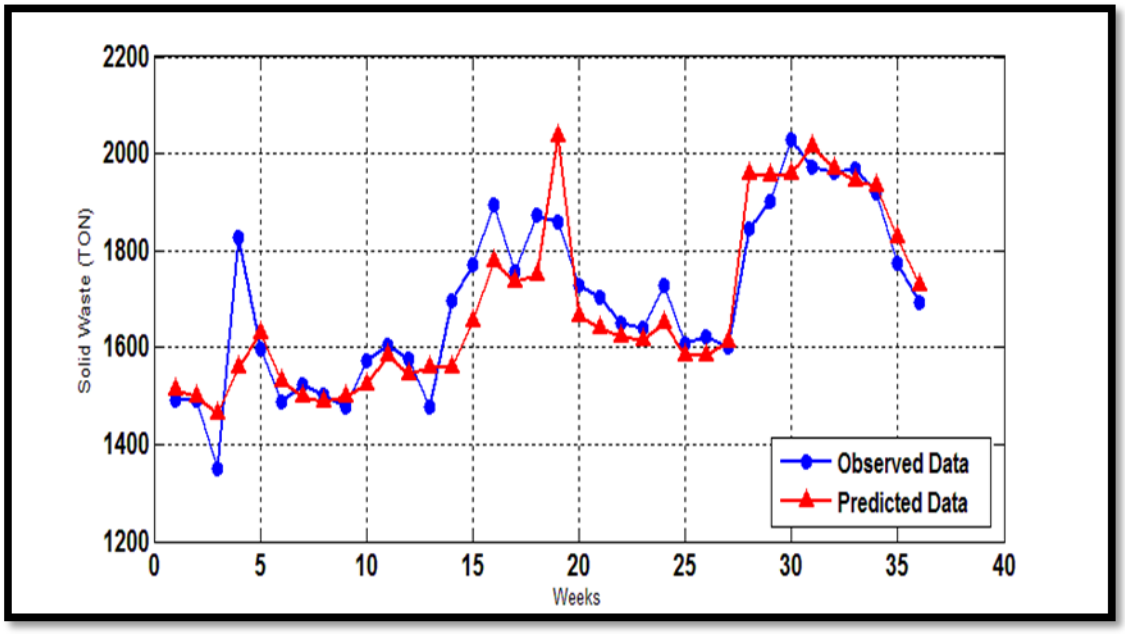


Figure 2. Observed amount of solid waste and predicted output of ANN Model with 1 neuron in hidden layer for testing data set.

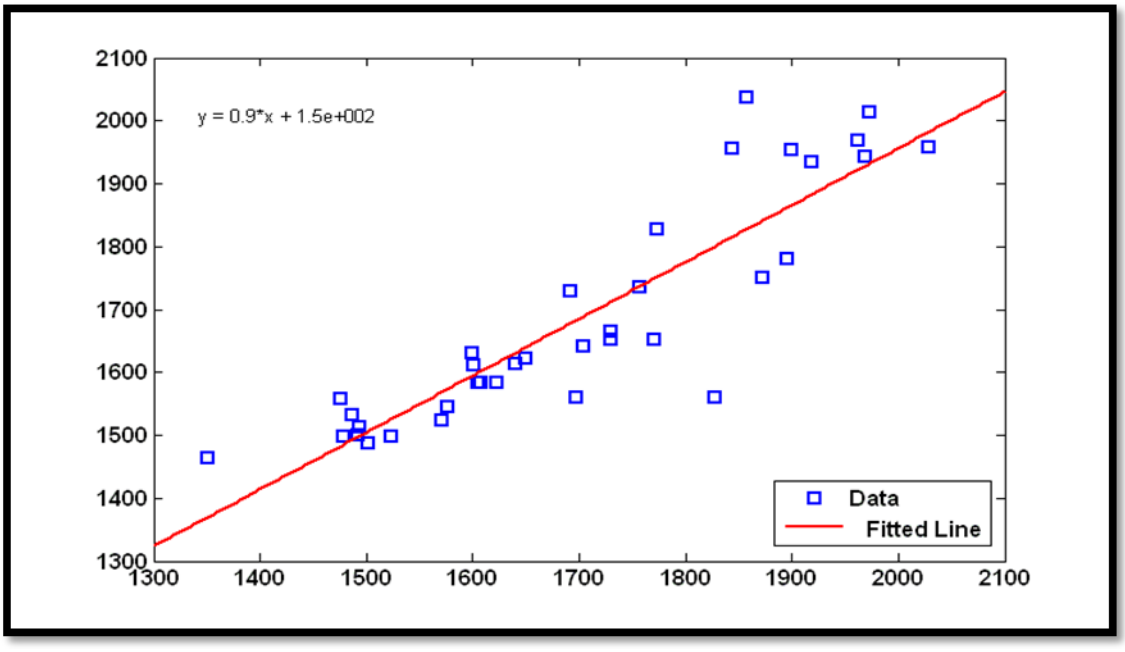


Figure 3. Scatter plot of predicted output of ANN Model with 1 neuron in hidden layer for testing data set versus observed amount of solid waste.

In the second step, a neural network with two neurons in hidden layer was applied. Training and testing data sets are applied to the ANN. The correlation coefficient (R2) of testing set increases from 0.992 to 0.989 and the correlation coefficient (R2) of training set increases from 0.975 to 0.989.

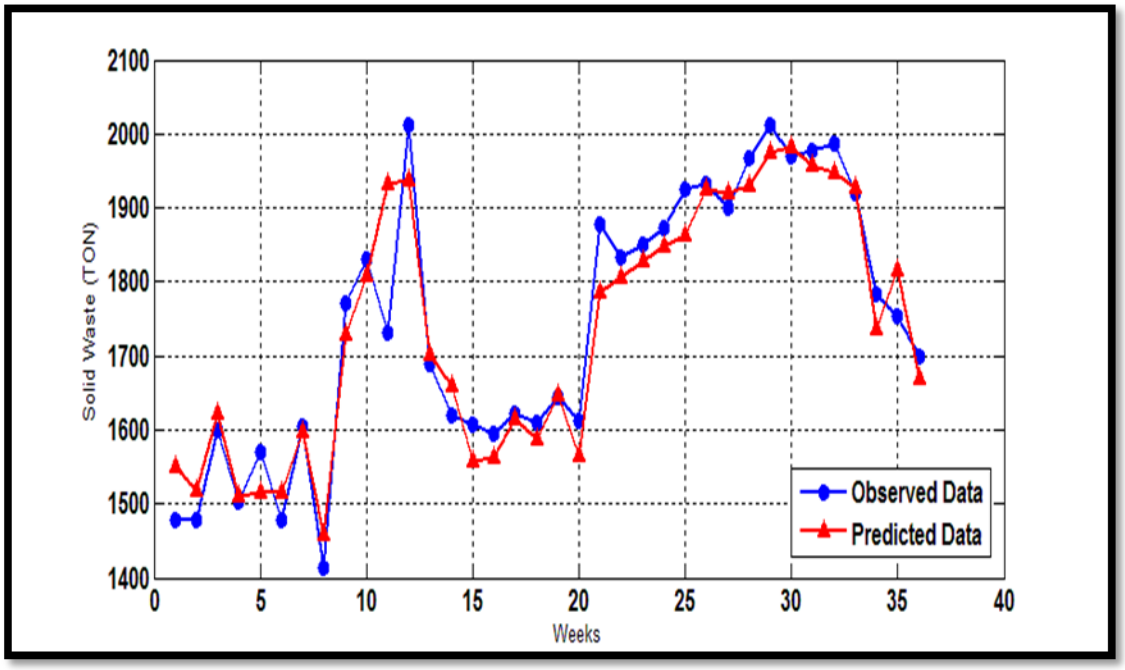


Figure 4. Observed amount of solid waste and predicted output of ANN Model with 2 neurons in hidden layer for testing data set.

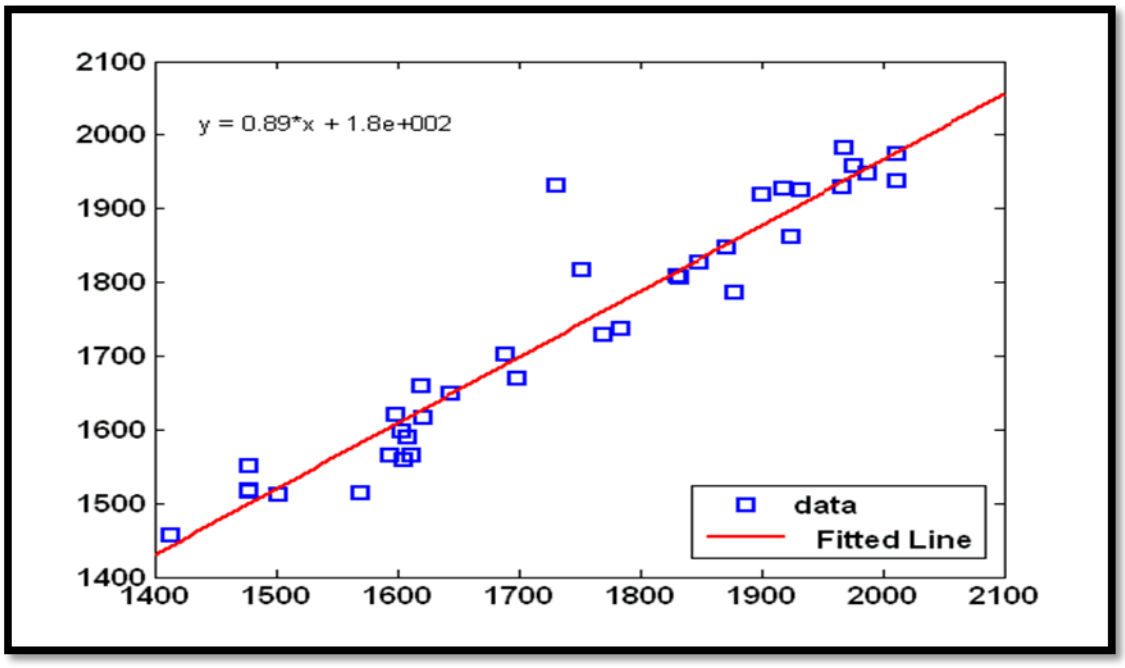


Figure 5. Scatter plot of predicted output of ANN Model with 2 neurons in hidden layer for testing data set versus observed amount of solid waste.

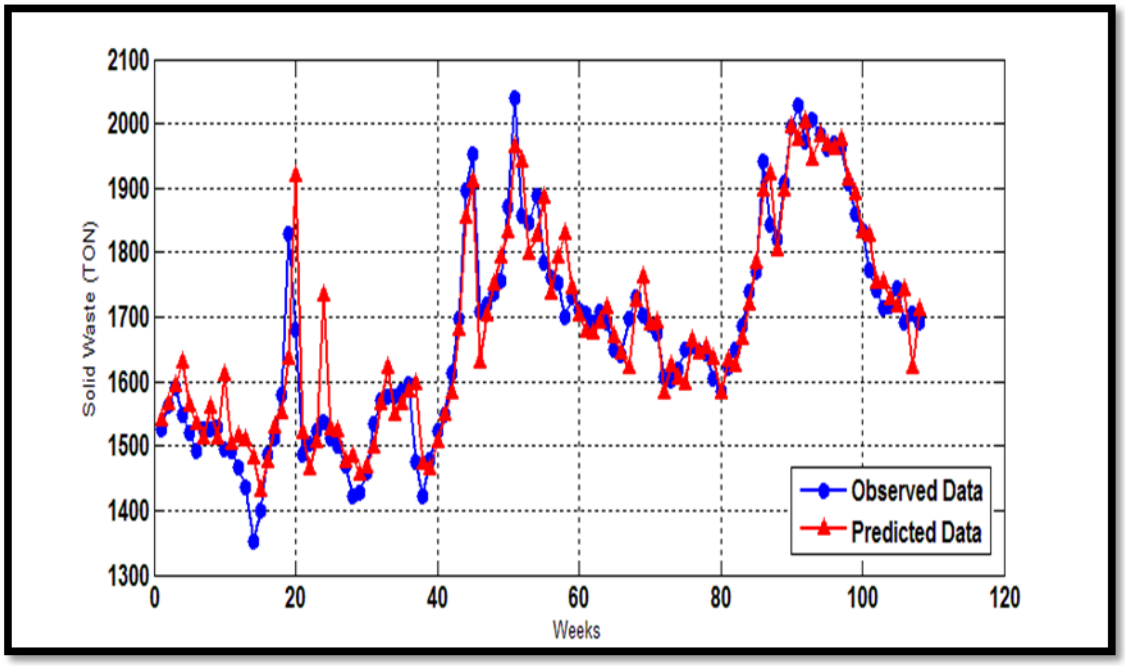


Figure 6. Observed amount of solid waste and predicted output of ANN Model with 2 neurons in hidden layer for training data set.

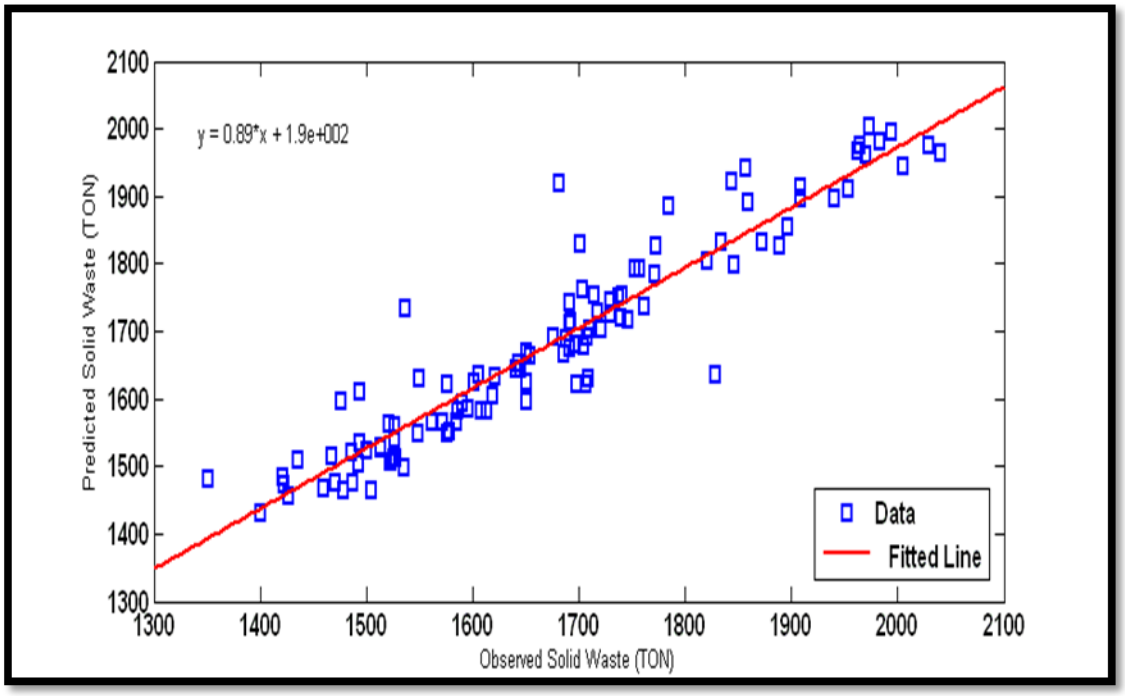


Figure 7. Scatter plot of predicted output of ANN Model with 2 neurons in hidden layer for training data set versus observed amount of solid waste.

5 Conclusion

Accurate prediction of solid waste generated is of vital importance in the municipal solid waste management. Thus, the goal of this study was to provide an appropriate model to predict this quantity. The most unique part of this model is that for the first time prediction of solid waste has been done by the analysis of artificial neural network and combining the amount of generated waste, fuel consumption, total of labor and quantity and quality of transport as input data. According to the results ANN structure with 2 neurons in hidden layers was selected as an optimal one which can minimize the calculation. The methodology or an adapted form of the methodology might also be applied to other fields, subject to a study of the requirements in each place. Therefore the goal of this research was offering a suitable model to predict this quantity and link to the geospatial environment.

At last with respect to MAE, MARE, RMSE and R2 suitable models were selected for the study after the mentioned model performing, correlation coefficient (R2) and mean absolute relative error (MARE) in neural network for test have been achieved equal to 0.989 and 0.06 respectively.

References

1. Adhikari, B.K., Barrington, S., Martinez, J., 2006. Predicted growth of world urban food waste and methane production. Waste Management Research 24, 421– 433.
2. Chi, Y., Wen, J. M., Zhang, D. P., Yan, J. H., Ni, M. J. And Cen, K. F., (2005). HCl emission characteristics and BP neural networks prediction in MSW/coal co-fired fluidized beds. J. Environ. Sci., 17, 699-704.
3. Dong, C., Jin, B. and Li, D. (2003). Predicting the heating value of MSW with a feed forward neural network.Waste Manag., 23, 103-106.
4. Jain, A. and Srinivasulu, (2006). Integrated approach to model decomposed flow hydrograph using artificial neural network and conceptual techniques. J. Hydro., 317, 291-306.
5. Jalili M, Noori R (2008). Prediction of municipal solid waste generation by use of artificial neural network: a case study of Mashhad. Inter J Env Rese, 2: 22-33.
6. Karaca, F. and Özkaya, B., (2006). NN-LEAP: A neural network-based model for controlling leachate flow-rate in a municipal solid waste landfill site. Environ. Model. Software, 21, 1190-1197.
7. Nadi B.,Ahmad ,R. ,A sharif,A noordin (2009)Use of geospatial technology for landfill site selection. Journal of Environment and engineering. 3 (9) USA.
8. Nadi, B., Mahmud, A. R., Ahmad, N., Farjad, B., Arvinpil, b., Amani, A., et al. (2010). Managing of Urban Solid Waste by Geoinformatics Technology. International Geoinformatics Research and Development Journal, 1(1), 70-80.
9. Nadi, B., & Shamshiry, E., & Rodzi, A. (2011). Response Surfaces Model For Optimization of Solid Waste Management. International Journal of Chemical Engineering and Applications (IJCEA); ISSN: 2010-0221. Vol.2, No.1.201.
10. Nadi, B., Mahmud, A. R.,(2010). Geospatial Modeling for Urban Solid Waste Management. ISBN 978- 3-639-28392-1. VDM Verlag Dr. Müller GmbH & Co. Germany.
11. Nasir, A.A., 2007. Institutionalizing Solid Waste Management in Malaysia: Department of National Solid Waste Management. Ministry of Housing and Local Government Malaysia, Power Point Presentation, December 6.
12. Noori R, Kerachian R, Khodadadi A, Shakibayinia A (2007). Assessment of importance of water quality monitoring stations using principal component and factor analyses: a case study of the Karoon River. J Water & Wastewater, 63: 60-9 (Language: Persian).
13. Noori R, Abdoli MA, Ameri A, Jalili- Ghazizade M (2008). Prediction of municipal solid waste generation with combination of support vector machine and principal component analysis: a case study of Mashhad. Environmental Progress, DOI 10.1002/ep.
14. Noori R, Abdoli MA, Farokhnia A, Abbasi M (2009). Results uncertainty of solid waste generation forecasting by hybrid of wavelet transform-ANFIS and wavelet transform- neural network. Expert Systems with Applications. DOI 10.1016/j. eswa. 2008.12.035.
15. Rigamonti, L., Grosso, M., Giugliano, M., 2009. Life cycle assessment for optimizing the level of separated collection in integrated MSW management systems. Waste Management 29 (2), 934–944.
16. Sahoo, G. B., Ray, C. and De Carlo, E. H., (2006). Use of neural network to predict flash flood and attendant water qualities of a mountainous stream on Oahu, Hawaii. J. Hydro., 327, 525-538.
17. Shamshiry,E., Nadi,B., Mokhtar ,M., Komoo,I., and Hashim,H., (2011). Urban solid waste management based ongeoinformatics technology. Journal of Public Health and Epidemiology Vol. 3(1), January 2011.ISSN 2141-2316 ©2010 Academic Journals.
18. Shrestha, S. and Kazama, F., (2007). Assessment of surface water quality using multivariate statistical techniques: A case study of the Fuji river basin, Japan. Environ. Model. Software., 22, 464-475.

19. Shu, H. Y., Lu, H. C., Fan, H. J., Chang, M. C. and Chen, J. C. (2006). Prediction for energy content of Taiwan municipal solid waste using multilayer perceptron neural networks. J. Air and Waste Manag. Assoc., 56, 852–858. M. Jalili Ghazi Zade and R. Noori 22.
20. Sujauddin, M., Huda, S.M.S., Rafiqul Hoque, A.T.M., 2008. Household solid waste characteristics and management in Chittagong, Bangladesh. Waste Management 28, 1688–1695.

An Autonomous Approach for Environmental Impact Assessment in Alexandria Marine Environment

Hesham A. Elzouka
Computer Engineering Department
Arab Academy for Science and Technology, Alexandria – Egypt
helzouka@aast.edu , helzouka@hotmail.com

Abstract

The objective of this paper is to propose a new autonomous model for the use of GPS devices that tightly coupled with GIS, to address the necessary framework and techniques needed to plan, design, and implement an assessment tool for solving problems associated with marine environment in Alexandria Bay. This work also aims at analyzing the data of the proposed system towards the elimination of oil pollution in the study area. In addition, the proposed information system will be used in monitoring, detecting and managing the oil spill crises in marine environment. Throughout the entire paper we assume the prior knowledge of location and movement of the spill which can be calculated from satellite remote sensing imagery using change detection techniques play a major role in determining the oil spill trajectory modeling for the study area.

Keywords: *GIS, Artificial Neural Network, Marine Environment, Genetic Algorithm, Modelling and Simulation*

1. Introduction

Geographic Information System (GIS) is used to utilize the core geographic nature of any information with the aid of image processing technologies used to describe several specific applications. These applications includes oil spill response, mapping historical change, evaluating sediments and water depths, non-point source pollution monitoring, surface runoff modeling, accuracy assessment, and many others.

Actually, GIS goes into depth on topics such as data structures and spatial analysis, focusing on global change research, land and water resource management, and environmental risk assessment. All in all, the application of geographic information systems to environmental, safety, and health (ES & H) information management can help any organization that deal with their ES & H information management need into the context of the overall corporate information flow, making information easily accessible, and transcending traditional facility management applications [1].

GIS coupled with intelligent predication system for analyzing the monitored spill trajectory such as the one proposed in this paper can help to solve such a problem. As a consequence, there is a strong demand for accurate and up-to-date spill information from satellite images plus other data resources to produce a map of affected areas. But before presenting our model, we must describe in detail the relevant parts of the oil/fuel and the impact it is having on the marine environment. The congestion levels that were studied were limited to three levels: free-flow, heavy and light spillage, which was enough and appropriate according to our case study.

The paper is organized as follows: In section 2 we will briefly mention the related work to our research and the environmental impact of oil pollution. In section 3 we will describe the proposed approach for the oil spill in Alexandria shoreline. Modeling of existing environment is analyzed in section 4. Finally, the oil spill trajectory and action planes are presented in section 5 and 6 respectively.

2. Related work

There are several collaborative efforts from private companies, government agencies, and research community to collect sea environmental data including location of sensitive areas. Ratchata and Wasan [2] found that most of these efforts are focused on the installation of intelligent multi-sensor system, such as intelligent temperature

sensors with image processing capability and inductive loops.
According to our survey, intelligent multi-sensor system can be classified into two types: fixed sensors and mobile sensors [3].
Fixed sensors have been widely used in many intelligent detection system (IDS) applications, and include loop-coils, radars and intelligent video cameras with image processing capability. However, fixed sensors are vulnerable to extreme weather events and climate change impacts. And the installation of fixed sensors to cover all sensitive areas in major marine environment is neither practical nor economically feasible.
On the other side, Mobile sensors are a low cost alternative that can be used as a complementary solution to fixed sensors for increasing coverage area without additional investment on expensive sensor infrastructure.
In addition, GPS-based sensors are far more efficient to pinpoint spillage locations, thus they can provide high accurate trajectory prediction information.
There have been several research proposals using GPS and cellular phone systems to estimate spill volume, movement times, wind, and current velocity. The possibility of using Cell Dwell time (CDT) to detect the spread of the oil spills was also tested. CDT is the time that a mobile phone attaches to a mobile phone service antenna to provide rough journey speed.
The GPS data would provide more precise positioning information than that roughly provided by the CDT .
Although, theses sensor infrastructures will provide accurate positioning information, very limited number of areas is equipped with them due to their high initial and maintenance costs.

2. Environmental Impact of Oil Pollution

Input of petroleum hydrocarbons into the marine environment ranges from diffuse chronic inputs such as terrestrial run-off and natural seeps to large point source releases as tanker spills for example. Deliberate release of oil into the world's oceans from marine operations or land-based activities is a relatively more important pollution source than accidents involving massive inputs of oil. In these days, researcher estimated that the amount of oil released into the Mediterranean was between 0.5 and 1 million tons per year, with half being discharges from the coast and half in the open sea.
Although, the amount of oil transported over the world's oceans has considerably increased, there has been a significant reduction of the quantity of oil discharged in the sea during transportation, due to the force of some conventions such as MARPOL convention [4].
It must be considered that in late 2009, 10 loading terminals out of the 19 existing in the Mediterranean did not possess deballasting installations, although they handled over 250 million tons of oil traffic.
There are more than 100 oil refineries located along the coasts of the Mediterranean. The input of petroleum hydrocarbon from these sources into the Mediterranean has been conservatively estimated at 30,000 tons per year.
The largest amount of this oil comes from the old refineries, which were not designed with water treatment as priority.
No figures or estimates are available regarding the amounts of petroleum hydrocarbons carried directly through land run-off into the Mediterranean or indirectly via rivers.

3. PROPOSED APPROACH

In our proposed approach we assume the prior knowledge of the location and the movement of the spill can be detected by GPS-based sensors or calculated from satellite remote sensing imagery using change detection techniques that is out of the scope for this paper.
Our proposed approach suggests the monitored spill trajectory and the nature of the spill oil to be selected by our model takes into consideration the knowledge of the sensitive areas and the provided risk map for Alexandria Bay.
The proposed approach makes use of the artificial neural network to classify the pixels of the input images into different pixel classes, based on color and texture features, to produce the best way of handling the spill under such circumstances.
A genetic algorithm based decision engine chooses the best solution to take based on the trajectory of the spill slick and its impact on the marine environment.

3.1 Data Collection

A notebook attached with a USB GPS device is used to collected date, time, latitude, longitude, wind-current direction, velocity and duration from interactive simulation GPS experiment. In the experiment, oil spill slicks were gathered from eleven locations along the Alexandria coastline [5]. The simulated full image consists of 800x800 pixels, with each pixel having a digital number representing its relative brightness and color. The computer treats each digital value received, in its proper category. Hence, the digital values virtually and meaningfully simulate the remote sensing data.

3.2 Artificial Neural Networks

Artificial Neural Networks (ANN) is a mathematical model that tries to simulate the structure and functionality of biological neural networks. It consists of an interconnected group of artificial neurons. ANN is an adaptive system that changes its structure based on information that flows through the network during the learning phase. In general, the architecture of a neural network as shown in figure 1 consists of three node layers: one input layer, one hidden layer, and one output layer [6]. All the nodes are fully connected as shown and the neural network model will adjust the weight of each node to reflect the patterns of the trained data.

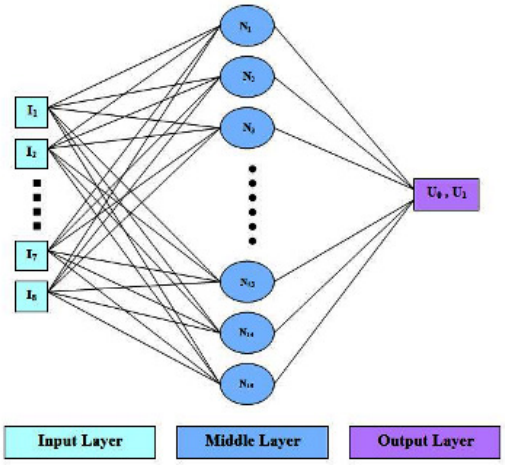


Figure 1. Artificial Neural Network

In the input layer of the ANN the following features used as input data: The day of the week, the time in minutes, the average speed of wind and current (km/h), the direction of wind and current, and the oil spill volume (tones/km^2).
The Oil spill thickness is classified in to one of the following three classes: Level 1; Light spillage which is kept within approximately 2 feet of the sea surface, then Level 2; Heavy spillage with 5 feet of the sea surface, and Level 3: Lubricating oil for more than 20 feet from the sea surface [7].
The output layer consisted of three neurons that is the targeted level 3, judged by participants. The neural network was trained using back propagation technique as illustrated in figure 2.

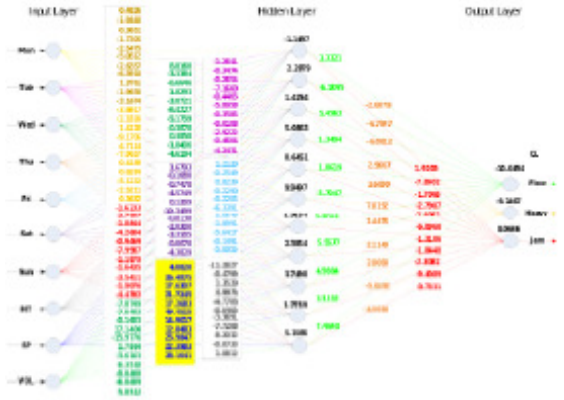


Figure 2. Trained Artificial Neural Network Configuration

3.3 A Genetic Algorithm Encoding

In this paper we assume that, the prior knowledge of the location and velocity of oil spills is "symbolic" and stored accordingly, which allow us to predict the trajectory of the oil on the sea using a genetic algorithm based decision engine. Genetic Algorithms (GAs) are adaptive heuristic search algorithms which are premised on the evolutionary ideas of natural selection and gene types [8]. The encoding system is represented as a chromosome which, in turn, is represented by an integer array. Thus, each oil slick index is a gene of the chromosome and each gene contains the number of variables associated with each oil slick, taking into consideration that each variable having a digital number representing its relative class and the impact on the environment. The computer then treats each digital value received, in its proper category. The number of variables for each slick/class is achieved from the simulated remote

sensing data as mentioned previously, where the fitness function F(x) is defined as:

$$F(x) = 1 / \text{Oil dispersion} + \text{environmental disasters}$$

According to the fitness function the two parents with best fitness are selected for mating and a combination of cut and splice and Multipoint crossover is used to breed new offsprings. Two selected parents are crossed over with gene crossover probability between 70% and 90%. Consequently, the algorithm will allow us to compute the impact of oil and related chemical spill over a marine environment.

In the following sections, genetic algorithm and GIS based solution system will be used simultaneously to simulate the dynamic phase behavior of the oil spill in the provided environment.

4. Modeling the Existing Environment:

The scope of this paper doesn't include an oil spill simulation trajectory run (model) only, but also determining portion of the shoreline to be protected in case of an oil spill. We have attempted to simulate the movement of an oil slick towards the shoreline. We used for this exercise ARC-INFO GIS software package [9], in order to model wind speed, direction data, water current direction and graphical representation of all these parameters. The base map used in the model was preexisting topographic map obtained from the Alexandria-Mediterranean research center [10]. Modeling information such as movement of the oil slick, distance and time for the spill to reach a certain point on the shoreline, the exact location of collision of the oil with the shore wasn't an easy task. It is properly assumed here that in case of spill proper monitoring and observation at sea will take place by satellite, helicopter or vessel based on the wind data available for a given time in the year. For example, in June, July and August it is clear that the winds are predominantly north western at an average speed of 10 knots reaching 20 knots [11] as shown in figure 3. The analysis shows that the North wind is prevailed in these months and it forms 84.9% of the total blowing wind, which is distributed with a percentage of 44.2% North of the NorthWest and 21.21% West of the NorthWest. In short, it means that the wind blowing from the NorthWest is equivalent to 63% on which it confines between angles 285° and 344° with a speed that differs from 1 to 10 knot/hr for 42.8 % and 20 knot/hr for the rest 20.5%. All of that North Sea wind ends up in the west of Alexandria. By adding the North wind which forms 14.3% of the blowing wind the total rate of the North wind reaches 77% while the East wind isn't more than 7.3%. The rate of changes in wind direction doesn't exceed 8% [12].

On the other hand, local water currents in and around Alexandria shoreline may have far greater control over a spill trajectory than the generalized water flow direction in the open sea, peculiarly those current in the west-east direction parallel to the shoreline. In addition, any spill scenario must take in consideration the oil volume, the ratios of spill, the area of spill, the thickness of spill and the evaporation, spreading and dispersion rates.

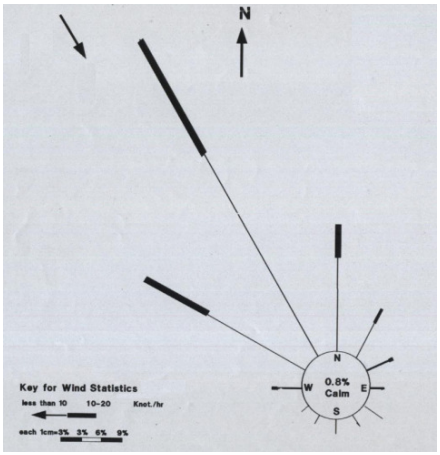


Figure 3 : Graphical Representation of wind, speed and direction

For the lowest of the proposed spills below 10 ton spill evaporation, dispersion and the break up in multiple patches under the action of waves would be its fate. However, for the largest of the area of 100 ton spill, it might have an impact on the shoreline if it reaches the coast and this would be under higher wind conditions [13].

In contrast, based on the above data, it is assumed here that if the spill reaches the shoreline it will affect the geographic zone and most visited beaches in Alexandria such as Montazah, Maamoura and Abu Qir as well as exposing the entire coast up to Rashid (Rosetta) to a minor risk of pollution.

Studding the water which is discharged from the rivers in the Mediterranean Sea constitutes, we

found that it represents only 1/4 of the Mediterranean water. Therefore, there is always a permanent stream of water which comes to the Mediterranean from the Atlantic Ocean through the Gibraltar Strait [14]. The GIS modeling system shows that the main part of that stream goes east in parallel to the North coast of Africa causing the Mediterranean current to affect the Mediterranean sea shore and the sandy patches and barriers stretched along Gulfs which spread on the Mediterranean coast.

On the other hand, a number of biological terms must be taken into consideration in case of modeling the coastal waters around Alexandria which contain fishing grounds and substantial bottom communities such as: types of fish, crustaceans, crabs, zooplanktons, marine turtles, oysters, clams, mussels, seahorses, birds and sea grass [15]. However, the trajectory model cannot simulate these areas in general in the case of an oil spill, due to their comprehensive coverage in these coastal areas.

5. Oil Spill Trajectory

In order to determine the portion of the shoreline to be protected in case of an oil spill, we have attempted to simulate the movement of an oil slick towards the shoreline as shown in figure 4.

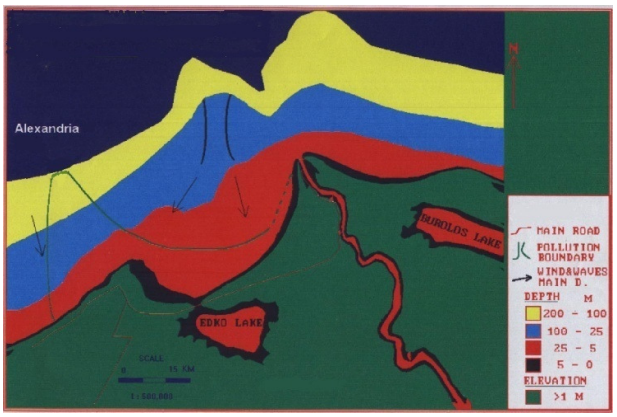


Figure 4: GIS Model for Pollution Impact Assessment

We have used oil spill simulation trajectory software to model and display wind speed and direction data, water current direction and graphical representation of all these parameters. The months of June, July, and August were used for the model as mentioned before. It is properly assumed here that in case of spill proper monitoring and observation at sea will take place by remote satellite images, helicopters or vessels.

Based on the wind data available for the month of July and August it is clear that the winds are predominantly northwesterly at an average speed of 10 knots reaching 20 knots.

Local water currents in and around the proposed studding area which is shown in figure 5, may have far greater control over a spill trajectory than the generalized water flow directions, particularly those currents in the west-east direction parallel to the shoreline. Other parameters such as the oil volume, the radius of spill, the area of spill, the thickness of spill and the evaporation, spreading and dispersion rates have been taken into consideration the proposed GIS scenario.

Figure 5: Sensitivity Map

The simulation results show that for the smallest of the proposed spills below 10 ton spill evaporation, dispersion and the break up in multiple patches under the action of waves would be its fate.

However, the analysis show that for the largest of the proposed spills in the area of 100 ton spill, it might have an impact on the shoreline if it reaches the coast and this would be under higher wind conditions.

Based on the simulation results, the analysis showed also that when the spill reaches the

shoreline it would affect the geographic zone between Montazah and the Abu Qir bay.

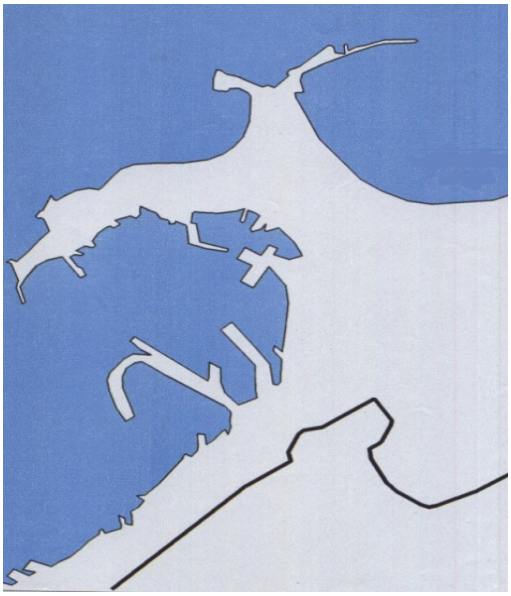


Figure 4: Eastern Port

Following these results, we have attempted to rank the zones which could be affected by an oil spill and therefore need protection. We have ranked them into "priorities" and according to the genetic algorithm, where priority A will be the most critical in descending order. Therefore, Priority A, zones are: Montazah(1), Maamoura (2), the area from ELMeadia to the town of Abo Qir east is the last one. One component of the wind and currents might lead a slick to the Eastern Harbor, which is therefore considered under priority A zoning as shown in figure 4. Priority B zones are from the El Meadia up to the west on the way to the Rosetta. These 2 divisions are based on the aesthetic, tourist, commercial and land use of the zones with special attention to fishing rounds. Also, environmental benefit analysis is considered as a vital element in the choice of zones to be protected.

6. Action Plans for Oil Spill trajectory

The Environmental law and its executive regulations, require operators to prepare an oil spill contingency plan when operating in the Egyptian coast line. In addition to the national law both the Barcelona and the OPRC conventions [16], signed by Egypt give detailed specifications to the establishment of contingency measures during any operations near the shore line. In the provided GIS model, the components of the strategic analysis are a risk analysis based on an integrated analysis of potential oil spill movements, mapping of ecological sensitive areas around the spill site and the risks for oil spills during operations. This analysis is illustrated in figure 5.

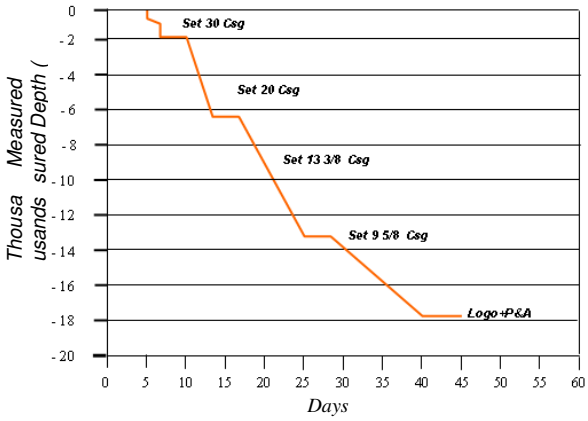


Figure 5: Risk Analysis

Following the outline of the potential risks and impacts of oil spills in the area, an action plan to be used in the case of oil spills is developed through the database associated with the GIS base model along with the applied ANN model. The model could be applied for the whole Mediterranean coastline of Egypt. Figure 6 shows the generalized environmental sensitivity features map for coastal areas of Egypt.

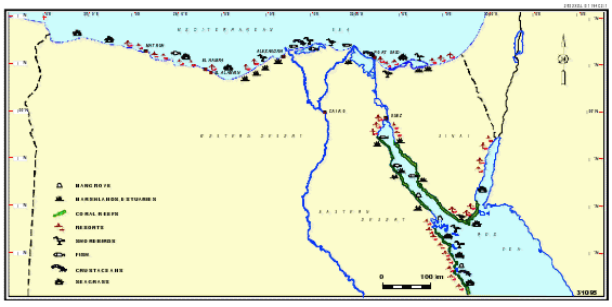


Figure 6. Sensitivity Map for Coastal Areas of Egypt

7. Conclusion and Discussion

In this paper we presented an oil spill simulation trajectory model for Alexandria shoreline to be protected in case of an oil spill. We have

simulated the movement of an oil slick towards the shoreline.
GIS was integrated with genetic algorithm model in order to display wind speed, water current direction and graphical representation and modeling all sensitivity maps.
The analysis showed that for lowest proposed spills below ten ton, the spill will be evaporated, dispersed and braked up into multiple patches under the action of waves.
However, for the largest simulated spills of one hundred ton, the spill will have an impact on the shoreline if it reaches the coast and this would be under high wind conditions.
Based on the sensitivity map and the data given for the provided area, we found that the spill will affect the geographic zone between Montazah and Abu Qir bay, only if the spill reaches the shoreline.
Following this assumption, we then attempted to rank the zones which could be affected by an oil spill using a fitness function. By the aid of GIS tools, we have ranked these zones into priorities, ranking from the critical zones, Abo Qir, Maamoura, and Montazah, till the Eastern Harbor of Alexandria.

8. References

1. Korte, G., The GIS hand book, Third edition. On Word Press Camino Entrada, USA , 1994.

2. Thammasak Thianniwet, and et al., Classification of Road Traffic Congestion Levels from GPS Data using a Decision Tree Algorithm and Sliding Windows, Proceedings of the World Congress on Engineering 2009 Vol I, 2009.

3. G. Rose, Mobile phones as traffic probes: Practices, prospects and issues, Transport Reviews, vol. 26, no. 3, p. 275291, 2006

4. International convention on oil pollution preparedness, response and cooperation, IMO, 1990.

5. NWRP, National water resources plan for Egypt, NWRP Decision support system, Technical report No. 4. 2008

6. H.C. Dai and C. Mcbeth, Effects of learning parameters on learning procedure and performance of a BPNN, Neural Network, 10(8), 1997

7. Oil Spill Prevention: A Proactive Approach, Issue paper, IOSC., 2003

8. Sachith Abeysundara, et al., A Genetic Algorithm Approach to Solve the Shortest Path Problem for Geographic Maps, Proceedings of the International Conference on Information and Automation, December 15-18, 2005, Colombo, Sri Lanka.

9. ESRI, ArchGIS 9.3.1 for dynamic map, http://www.esri.com/software /arcgis., 2008

10. http://www.bibalex.org/alexmed/., 2009

11. Yasser A., Guenet M., The use of Remote Sensing and GIS in the assessment of visual attributes: Case study of the northwestern coastal zone of Egypt. Faculty of Environmental Planning, University of Montreal, 1990

12. National Oil Spill Contingency Plan Egypt, EEAA, 1998

13. Baker, J.M., et al., Long term fate and effects of untreated thick oil deposits on salt marshes., Proceeding of the Oil Spill Conference, American Petroleum Institue, Washington, DS., 1993

14. Barcelona Convention, Convention for the Protection of the Marine Environment and Coastal Region of the Mediterranean, Barcelona, Spain, June 1995.

15. Aboul Dahab, et al., Chromium Biogeochemical Cycle in Abu Kir Bay, east of Alexandria, Egypt, Estuarine, Coastal and Shelf Science, Vol 29, Issue 4, October 2004.

16. International Convention on Oil Pollution Preparedness, Response and Co-operation (OPRC), London, 1990, Ratified in 1999.

Detection and Grading of Astrocytoma tumor in MR Brain Images using Neural Network

Ashwani Kumar Grain
E & Ec., PEC University of Technology
Chandigarh, India-160012
Email address: grain.office@gmail.com

Virender Rihani
E & Ec., PEC University of Technology
Chandigarh, India-160012
Email address: v.rihani@yahoo.com

Abstract: **The paper introduces an efficient method for detection and grading of Astrocytoma, a type of brain tumor in MR Brain Images. Gray Level Co-occurrence Matrix (GLCM) is applied to Image to get information like entropy, energy, contrast, correlation and ratio of brain tumor to the MR Image etc. These are used for creation of knowledge base. After training of ANN with input and target matrix, the ANN classifies type of brain tumor. These are classified as Grade I, II, III & IV Astrocytoma brain tumor as per WHO norms.**

Keywords: Grading of Astrocytoma Tumor, Artificial Neural Network, Back Propagation Algorithm, Gray Level Co-occurrence Matrix (GLCM), Confusion Matrix (CM).

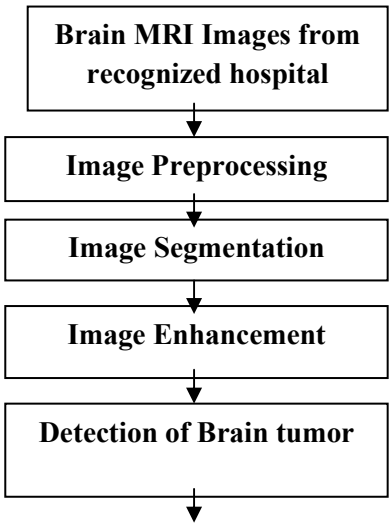
I. INTRODUCTION

Astrocytoma, a type of brain tumors dominates [3], all other types that occur in brain. This type of tumor is most lethal and very difficult to treat [1]. Astrocytoma tumors originate in a particular kind of glial-cells (The star-shaped brain cells) called astrocytes [16]. This type of tumor doesn't usually spread outside the brain and spinal cord and also it doesn't usually affect other organs. Astrocytomas are broadly of two classes: those with narrow zones of infiltration and those with diffuse zones of infiltration. According to WHO, all Brain tumors can be graded into four grades known as Grade I, II, III and IV [1, 3]. And are also classified as benign and malignant. Glioblastoma Multiforme (GBM) is most aggressive astrocytoma tumor and it belongs to grade-IV followed by Anaplastic Astrocytoma, which is classified as grade-III [3]. Grade I and Grade II are less common. These consist of relatively slow-growing astrocytomas as compared to Grade III & IV and are observed in the age of 0-5 years. Brain tumors are leading cause of cancer deaths in children under the age of 20 and are the second leading cause of cancer among 20-29 year old males [4]. Metastatic brain tumors result from cancers that spread from other parts of the body. 10-15% of persons suffering from cancer develop metastatic brain tumor [7]. Therefore safer methods have to be developed for overcoming them [1]. In recent years health sector has observed a dynamic growth in research works conducted in the area of MR Brain Images [1, 2, and 5]. MR images provide very clear data of human tissues, nerves and inside of the human body than any other method [3, 4 and 6]. Scientists have yet to find the root cause of

cancers. This paper provides an efficient solution to predict and classify the astrocytoma tumor using MR brain images. Artificial Neural Network is used for pattern recognition and multivariate analysis of MR image data [10, 15]. Feed-forward back-propagation algorithm with one hidden layer has been used to train the Neural Network.

II. METHODOLOGY

Images are segmented, features extracted from MR brain image, and knowledge base is created. Artificial Neural Network is used for pattern recognition and classification of brain MR images. Images are converted from any image format to gray or binary scale image, and enhanced using histogram equalization to reduce the possibility of any confusion. Features have been extracted using GLCM. A knowledge base is created which is used for, training ANN for pattern recognition, prediction and grading of unknown MRI brain image as per WHO norms. The trained ANN model is used to predict and grade of unknown MR images.



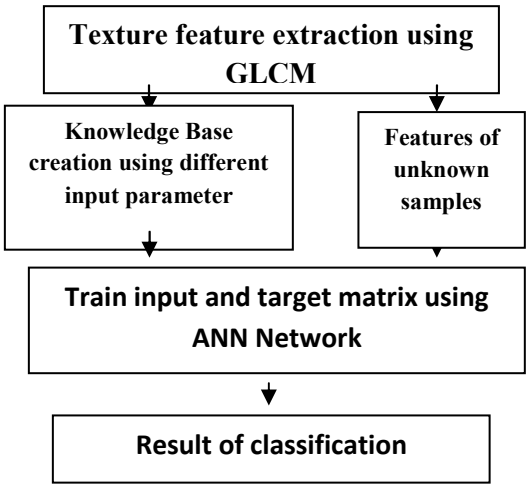


Figure 1. Block Diagram of the System

The steps used in the designed model are depicted in Fig 1.

Step1. Preprocessing of MR brain images [3, 4].

Step2. Edge detection using segmentation to locate tumor [3, 7, 11].

Step3. Features are extracted to reduce the original data set. Features used in this model are energy, correlation, contrast and homogeneity by Gray Level Co-occurrence method [1, 12,].

Step4. The input matrix consists of parameters based on ROI (Region of interest) [8, 13, 14].

1 Age
2 Genders
3 Seizures
4 Headaches
5 Progressive memory losses
6 Personality and behavior change
7 Visual problems
8 Malignant tumors
9 MRI taken in axial position
10 MRI taken in Saggital position

11 MRI taken in Coronal position
12 Edge clearly distinguishable
13 Metastasis tumors
14 Tumor is sparsely distributed
15 Tumor position is in frontal lobe
16 Tumor is positioned in Temporal lobe
17 Tumor is positioned in Parietal lobe
18 Tumor is positioned in cerebellum region
19 Tumor is positioned in Pineal region
20 Tumor shape is irregular in MRI
21 Multi-centric
22 Intracranial tumor
23 Non enhancing
24 Centered around thalamus or hypothalamus region
25 Contrasts
26 Correlation
27 Energy
28 Homogeneity
29 Entropy
30 Tumor Length ratio (horizontally)
31Tumor Length ratio (vertically)

Expected responses from the knowledge matrix on query are:

1 Astrocytoma (Grade –I)
2 Astrocytoma (Grade –II)
3 Anaplastic astrocytoma (Grade –III)
4 Gliobalstoma Multi Forme Astrocytoma (Grade –IV)
5 Oligodendroglioma (Grade-III) not an astrocytoma
6 Having unknown tumor other than astrocytoma like Anaoligodendroglioma.

Step5. Feed forward Back propagation

ANN is used as pattern recognition tool, which randomly selects samples for training, validation and testing. Training automatically stops when generalization stops improving, Mean Square Error (MSE) of the validation is the average squared difference between outputs and targets required. Lower values are better and zero means no error.

Step6. After successful training of model the test results are observed in the form of Confusion Matrix (CM), Mean Squared Error, and Percent Error & ROC plot. Percent Error indicates the fraction of samples, which are misclassified. A value of 0 means no misclassifications, 100 indicates maximum.

III TESTING AND RESULTS

MATLAB with neural network toolbox has been used for recognition. The best results were observed in a neural network, which has 31 input neurons, 6 output neurons and one hidden layer with 18 hidden neurons. After successful training, results are presented in form of CM and ROC plot in figure 2 & 3 respectively. Out of 49 Astrocytoma disease samples, three sample sets were randomly selected for training, validation & testing. The results of three sets (set A, set B and set C) of varying sample sizes for training; validation and testing are presented in table 1. Training and Validation are 100% successful for every sample set A, B and C. The results in table also show that testing is more than 85 % successful in each sample set A, B and C, which is much better result in comparison to experienced doctor. For each testing sample set A, B & C, correctly classified astrocytoma tumor with different grade as per WHO norm has been shown in table2. The results in table 2, show that training and validation phases have been 100% successful. Percentage Error for both testing and validation is

zero, which means there is no error. Testing phase has been 80% successful

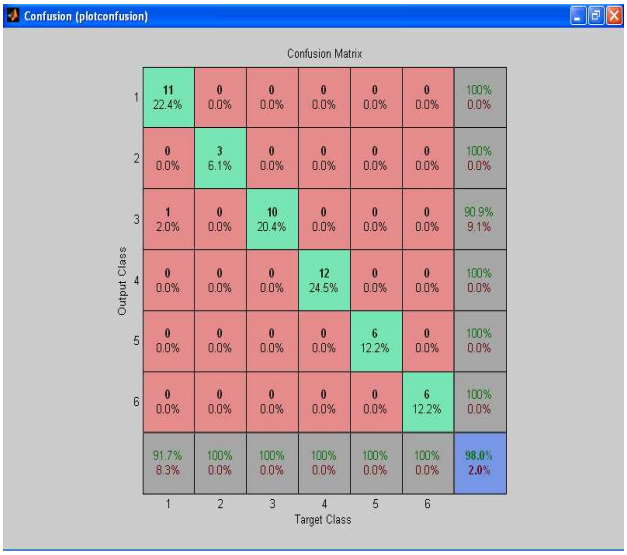


Figure 2. Confusion Matrix (CM)

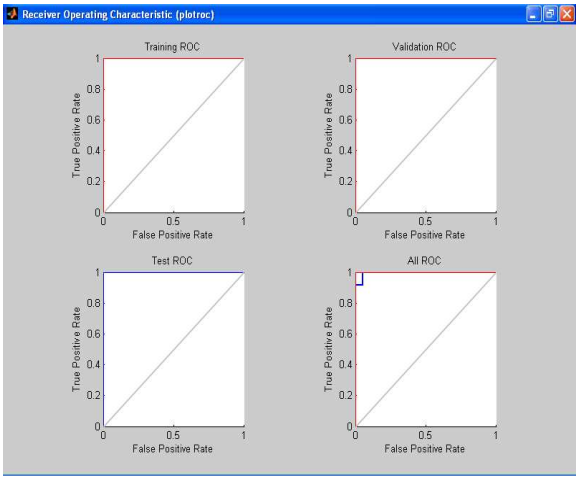


Figure 3. ROC plot

100 samples (Patient having different Brain tumors) have been used for ANN training. The results have also been obtained by varying the number of neurons in hidden layer and are presented in Table 3.

Table 1. Correct rate, % Error and MSE on optimized network.

No of samples taken for	Target	Output	Correct rate	MSE	% Error
Training	35	35	100 %	3.16e-7	0
Validation	7	7	100%	1.26e-3	0
Testing	7	6	85.7%	2.04e-2	14.3e-0
Training	30	30	100%	6.15e-7	0
Validation	7	7	100%	1.19e-3	0
Testing	12	11	91.7%	2.58e-2	8.33e-0
Training	29	29	100%	2.87e-7	0
Validation	10	10	100%	1.28e-2	0
Testing	10	9	90%	2.39e-2	10.0e-0

Table 2. Classification of testing samples of diff. astrocytoma tumor

Testing samples		**Astrocytoma tumor grading**					
		Grade I	Grade II	Grade III	Grade IV	Grade III not an astrocytoma	Other than astrocytoma
Set A	Target	1	0	3	1	1	1
	Output	1	0	2	1	1	1
	Correct classified	100 %	100 %	66.7 %	100 %	100 %	100 %
Set B	Target	3	1	2	5	1	0
	Output	3	1	2	4	1	0
	Correct classified	100 %	100 %	100 %	80 %	100 %	100 %
Set C	Target	5	0	4	0	0	1
	Output	5	0	3	0	0	1
	Correct classified	100 %	100 %	75 %	100 %	100 %	100 %

IV INTERPRETATION

Confusion Matrix: Confusion matrix contains information about known class labels and predicted class labels. The observations with known class labels are usually called the training data. As example (I, J) element in the confusion matrix is the number of samples whose known class label is class I and whose predicted class is J. Fig. 2 presents a confusion matrix for 49 samples, in which diagonal elements represent correctly classified samples. One sample which was targeted to first column, first row has after training has been shown to belong to first column third row. It means this sample was targeted to grade I but after training it is realized that this sample belongs to grade III (Anaplastic astrocytoma).

ROC (Receiver operating characteristic): It is a metric used to check the quality of classifiers. For each class of a classifier, ROC applies threshold values across the interval [0,1] to outputs. For each threshold, two values are calculated, the True Positive Ratio (sensitivity), and the False Positive Ratio (Specificity). The plotroc plots the receiver-operating characteristic for each output class. A better classification is achieved when line touches the upper-left edge of plot. In figure 3, training ROC, validation ROC, testing ROC lines hugs to upper-left corner with 100% sensitivity and 100% specificity. But in all ROC of ROC plot have 90% sensitivity with 90 % specificity.

GLCM (Gray level Co-occurrence Matrix) is a statistical method for texture analysis that considers the spatial relationship of pixels in image. GLCM created by calculating spatial relation of a pixel, with the intensity (gray-level) value *I*, with a pixel with the value J. Properties of GLCM are calculated by graycoprops function which includes contrast, correlation, energy and homogeneity. Entropy is a measure of randomness and characterizes the texture of input image.

V CONCLUSION

In this paper a novel method and an algorithm has been used to classify different Astrocytoma brain tumor. Using it one can predict and classify MR image of patient's brain into Grade I, Grade II, Grade III and Grade IV Astrocytoma as per WHO norm. The result obtained using ANN provide supportive role to the Biopsy method.

VI ACKNOWLDGEMENT

The authors acknowledge with thanks Dr. Deepak Tyagi., Neurosurgeon (NIMS hospital, Chandigarh, India) for providing the real MRI data and guidance used in this study.

VII REFERENCES

[1] 'D.Jude Hemanth, C.Kezi Selva Vijila and J. Anitha', "Comparative Analysis of Neural Model for MR Brain Tumor Image Segmentation", World Congress on Nature & Biologically inspired Computing, pp. 1616-1619, 2009.

[2] 'Jesus J.Caban, Noah Lee, Shahram Ebadollahi, Andrew F.Laine and John R.Kender', "Concept Detection Longitudinal Brain MR Images Using Multi Modal Cues", IEEE transaction on MRI Brain Images, pp. 418-421, 2009.

[3] 'Dipali M. Joshi, Dr. N.K.Rana, V.M.Misra', "Classification of Brain Using Artificial Neural Network", 2nd International Conference on Electronic Computer Technology, pp. 112-116, 2010.
[4] 'Ahmed Kharrat, Mohamed Ben Messaoud', "Detection of Brain tumor in medical Images", International Conference on Signal and systems, pp. 1-6, 2009.
[5] 'D.Glostos, P.Spyridonos, P.Petalas, D.Cavouras, V.Zolota, P.Dadioti, I.Lekka, G. Nikiforidis', "A hierarchical decision tree classification scheme for brain tumor astrocytoma grading using Support Vector Machines", Proceedings of 3rd International Symposium on Signal Processing and Analysis, vol. 2, pp. 1034-1038, 2003.
[6] 'D.Selvathi, R.S. Ram Prakash, Dr. S.Thamarai Selvi', "Performance Evaluation of Kernel based techniques for brain MRI Data Classification" International Conference on Computational Intelligence and Multimedia Application, vol. 2, pp. 456-460, 2007.
[7] 'D.Jude Hemanth, D.Selevathi and J.Anitha', "Effective Fuzzy Clustering Algorithm for Abnormal MR Brain Image Segmentation", IEEE International Advance Computing Conference, Patiala, India, pp. 609-614, 2009.
[8] 'Chun-Yi Lin, Jun-Xun Yin, Li-hong Ma, Jian-Yu Chen', "Fuzzy Bayesian Network-based Inference in Predicting Astrocytoma malignant Degree," Proceeding of the 6th World Congress on Intelligent control and Automation, Dallan, China, pp. 10251-10255, 2006.
[9] 'Syoji Kobashi, Mieko Matusi, Noriko Inoue, Katsuya Kondo, Tohru Sawada, Yukata Hata', "Adaptive Brain Tissue Classification with Fuzzy Spatial Modeling in 3T IR-FSPGR MR Images", World Automation Congress, Budapest, Hungary, pp. 1-6, 2006.
[10] 'Shi Lei, Wang Xing-cheng', "Artificial Neural Network: Current Application in Modern Medicine", International Conference on Computer and Communication in agriculture Engineering, vol. 2, pp. 383-387, 2010.
[11] 'Nan Zhang, Su Ruan, Stephane Lebonvaller, Qingmin Liao, Yuemin Zhu', "Multi-Kernel SVM Based Classification For Tumor Segmentation of MRI Multi-Sequence", IEEE transaction on Brain Tumor classification, 2009, pp. 3373-3376.
[12] 'Fei Peng, Kehong Yuan, Shu Feng, Wufan chen', "Region feature extraction of brain CT image for classification", IEEE transaction on brain Image Classification, pp. 2495-2498, 2008.
[13] 'Evangelia I.Zacharaki, Sumei Wang, Sanjeev Chawla, Dong Soo Yoo, Ronald Wolf, Elias R. Melhen, Christos Davatzikos, "MRI-Based Classification of Brain Tumor Type and Grading Using SVM-RFE", IEEE transactions on Brain tumor classification, pp. 1035-1038, 2009.
[14] 'Jan Luts, Johan A.K.Suykens, Sabine Van Huffel, Teresa Laudadio, Sofie Van Cauter, Uwe Himmelreich, Enrique Molla, Jose Piquer, M.Carmen Martinez-Bisbal, Bernardo Celda', "Differentiation between brain metastases and glioblastoma multiforme based on MRI, MRS and MRSI", IEEE transaction on Brain MRI classification, pp. 1-8, 2009.
[15] 'Sergiy Stepanyuk', "Neural Network Information technologies of Pattern Recognition", MEMSTECH'2010, Polyana-Svalyava (Zakarpattya), Ukraine, pp. 210-216, 2010.
[16] 'Alex F. Johnson, Barbara Holcomb Jacobson', "Medical speech-language pathology: a practitioner's guide" Keystarter, Fall 97 II.

Integrated Controller For Fixed Speed, Grid Connected Wind Turbine, Based On Neural Networks.

A. Hashad[1], F.Z.Amer[2], A.M. El-Garhy[2], A.E.Youssef[3], S.M. Aly[4]
1. Al-naeim Consultancy, Qassim University – Buraydah. KSA.
2.Control Engineering Department. Helwan University. Cairo, Egypt.
3. Information systems Department, King Saud University. Ryadh KSA.
4. Wind Power studies department New and Renewable Energy Authority. Cairo, Egypt.

Abstract: *Electrical power production is the main target required from wind turbines. This paper describes an approach for wind turbine controller as a vital part of the turbine, where this controller is based on Artificial Neural Network technique and a control scheme has been applied and validated by detailed simulation in MATLAB 6.5/Simulink. The proposed controller enhances the reactive power affected by electrical grid voltage and/or load disturbances where the controlled variables for the controller are system voltage and power production. Final results were compared with practical database of wind turbine runs by conventional controllers without applying ANN, and found positive. The controller model is useful for wind energy developers for designing wind energy conversion systems, where ANN based controller is much faster and adaptive to maintain maximum power conversion efficiency which appears steady at maximum in the same area of the power/wind speed curve during load or wind sudden variances.*

Key words: generator, Capacity factor, Artificial Neural Network, simulation, wind turbine controller, voltage stability.

1. Introduction:

Control system in wind turbines plays important role to guarantee "Fail Safe" operation of the machine; this is done by integrating machine stability with utilizing the machine design for maximum power production. Wind Power Conversion depends essentially on the power coefficient, "Cp" of the machine which translates the efficiency of converting wind power to electrical power, where it is the ratio of power produced by a wind turbine to the power in a reference area of free wind stream, as will be explained later.

From this pivotal point any control system starts to be designed, and first portion to be studied in the system is the generator.

Conventional controllers uses all possible parameters to guarantee perfect aerodynamic power control which is definitely needed at variable wind speeds to limit the input power to the wind turbine, but in case of applying ANN technique, the controller can handle any variances or disturbances in operational parameters in shorter time to reach steady state case, also a turbine can continue in production mode without disconnecting from grid as currently happens in practical.

This approach differs than conventional controlling in focusing on voltage stability towards supply variances and load fluctuations under continuous connection to power grid. Where "ANN" is used in the controller structure to support the required "learning" for grid supply state to guarantee perfect capacitance bank value required to balance the system voltage against any turbulences of grid supply voltage or load unbalances. This is done by perfect exploitation of capacitance stages in reactive power module, resulting consequent improve in power production stability. [1-3].

For steady-state calculations of wind turbine, mechanical power can be determined by [4].:

$$Pm = \frac{1}{2}\rho ArCp(1/\omega)v^3 \qquad (1)$$

Where "ω" is the rotational speed of wind turbine blades, and it is called rotor speed, this rotational movement is transported to a gear box through a low-speed shaft. Then the speed is converted to actual rotational speed feeding the generator by high speed shaft from the other side of the gearbox, this speed is the synchronous speed "ω_s" once the generator is connected to grid. "ν" is the wind speed, "ρ" is the air density and "A_r" is the area swept by rotor. General behavior of a wind energy conversion system can be described in two main relations, power coefficient as a function of tip speed ratio "μ" and mechanical power as a function of wind speed, as shown in Fig. 1, where μ is defined as the ratio between the rotational speed of the blade tip and the actual velocity of the wind. [1].

Section 2 of this paper, explains the Self Exited Induction Generator "SEIG" design, in section 3, an abbreviated explanation for conventional PI controller is mentioned. Section 4, explains the proposed ANN controller, section 5 Shows the conclusion & finally the actual results of this integrated system compared with on land running turbines in Zafarana wind farm on red sea coast which currently contains more than 400 fixed speed, grid connected wind turbines, is presented in section 6.

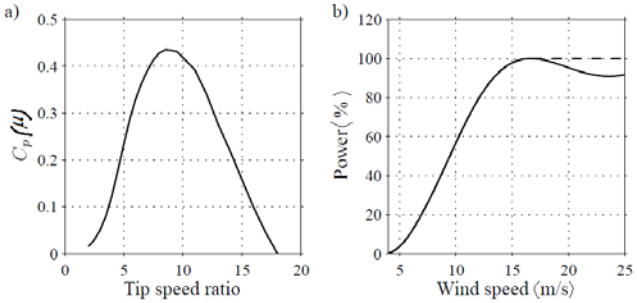


Fig. 1

2. Grid Connected SEIG Dynamic Modeling:

Self Excited Induction Generators are used for autonomous applications especially in wind power systems. The lower unit cost, brushless cage rotor

construction, absence of a separate DC source and better transient are its main advantages over the conventional alternators.

In this approach, the control is applied on SEIG, where an appropriate capacitor bank is connected across supply terminals of the generator by means of ANN decision. This helps to achieve balance on voltage and current components by compensating the VAR demand for the machine in good timing, but the important difference in this case "comparing with conventional controllers" is the fast and adapted response by ANN to maintain the stability to the turbine operation.

Parameters of this research case study machine are as per Vestas fixed speed, grid connected wind turbine: V60/660 KW: Generator rated voltage is 690 V/560A, 50 HZ working frequency, generator stator resistance Rs=0.0048Ω, Rotor resistance Rr=0.0087Ω, rotor reactance Xr=0.0897Ω, and stator reactance Xs=0.068Ω.

Modeling for the induction generator under study was presented in direct/quadrature reference, "d-q form", where it is easier to get the complete solution, transient and steady state of the self-excitation by this presentation. The following mathematical differential equations describe the generator by applying Kirchhoff's voltage law to the equivalent circuits of the SEIG as shown in Fig. 2; a) in quadrature axis and b) in direct Axis, we get the following [1],[5],[6]:

2.1 Stator Voltage:

Despite SEIG is preferred to be used in non conventional power conversions, it has also considered being variable voltage, variable frequency source, so in order to handle the power utility connection, it is necessary to guarantee stable and steady voltage and frequency of the system. Voltage can be driven as:

$$Vds = -Rs * Ids - (\frac{\omega}{\omega b})\varphi qs + \frac{d}{dt}(\frac{\varphi ds}{\omega b}) \quad (2)$$

$$Vqs = -Rs * Iqs - (\frac{\omega}{\omega b})\varphi ds + \frac{d}{dt}(\frac{\varphi qs}{\omega b}) \quad (3)$$

Where V_{ds}, V_{qs} are stator voltage in direct and quadrate axis respectively (Volt). And I_{ds}, I_{qs} are stator current in direct and quadrature axis respectively (Ampere). ω, is the rotor speed (rpm) and ωb is the base velocity ($\omega b = 2\pi f$, and equals to 314.16 as per 50 Hz working frequency).

2.2 Flux Linkage:

The initiation of the self excitation process is a transient phenomenon and it is better understood if analyzed using instantaneous values of current and voltage in d-q axis model to investigate the process of current and voltage build up during self excitation, where the main variables of the machine in rotating frame are flux linkages and load perturbations [1],[5].

Regarding the load current and the relation with the capacitor bank value, the following equations will clarify this:

$$\varphi ds = -Xls * Ids + Xm(Idr - Ids) \quad (4)$$

$$\varphi qs = -Xls * Iqs + Xm(Iqr - Iqs) \quad (5)$$

Where: φ_{ds}, φ_{qs} are stator flux linkage in direct and quadrant axis respectively (Weber), φ_{dr}, φ_{qr} are rotor flux linkage in direct and quadrant axis respectively (Weber), Xls is the stator reactance (Ohm) and X_m is magnetization reactance (Ohm).

2.3 Current & Capacitance:

$$Ids = \left[C * \left(\frac{d}{dt}(Vds) \right) + \left(\frac{Vds - \left(Xl * \frac{d}{dt}(ILds) \right)}{RL} \right) \right] \quad (6)$$

$$Iqs = \left[C * \left(\frac{d}{dt}(Vqs) \right) + \left(\frac{Vqs - \left(Xl * \frac{d}{dt}(ILqs) \right)}{RL} \right) \right] \quad (7)$$

$$ILds = \int_0^{600} \frac{Vds - (RL * ILds)}{XL} \quad (8)$$

$$ILqs = \int_0^{600} \frac{Vqs - (RL * ILqs)}{XL} \quad (7)$$

$$\mathrm{Im} = \sqrt{\left((Ids - Iqs)^2 + (Idr - Ids)^2 \right)} \quad (10)$$

Where, RL is load resistance (Ohm), IL_{ds} and IL_{qs} are load current in direct and quadrature axis respectively (Ampere) and I_m is the mutual Current (Ampere).

The capacitor bank consists of 4 stages, the effective capacitance C; is determined according to the reactive power produced and the load estimated on the output terminals and its value to be applied by the controller, [2],[3],[7].:

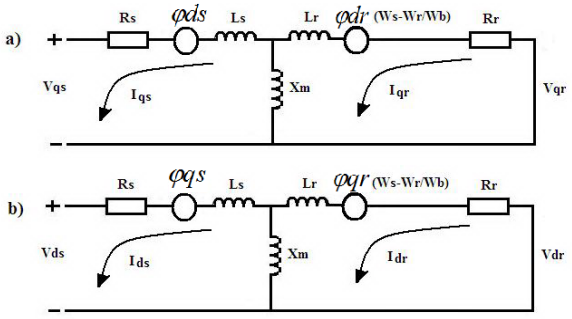


Fig.2

$$C = \frac{C\max}{\left((1-\lambda)^2 + \delta\lambda^2 \right)} \quad (11)$$

Where, δ is the ratio between maximum to minimum value of capacitor bank and λ is the duty cycle value which is the portion of time during which ANN switches from ON state to OFF state or switches between capacitance stages. [1],[2].

2.4 Power & Torque Presentation:

Aerodynamic lift and drag force on the rotor blades, produces the torque on the turbine rotor, the mechanical power produced by a rotor is purely a function of the geometry and the incident velocity. The C_P for any fixed rotor geometry is a well-prescribed function of the blade tip speed ratio with a single maximum value. The torque produced by the rotor can be controlled by varying the blade pitch angle; which is defined as the deviation angle of the blade away from its longitudinal axis. This is translated by the following equations [7],[6].:

$$Tm = Pm/\omega b \quad \& \qquad (12)$$

$$Te = (\varphi ds * Iqs) - (\varphi qs * Ids) \qquad (13)$$

Where, T_m and T_e are mechanical and electrical Torque respectively. Basically, rotor rotation is a balance between the aerodynamic torque applied by the wind and the electrical torque applied by the generator. The power coefficient is a measure of the mechanical power delivered by the rotor to the turbine's low speed shaft. It is frequently defined as the ratio of the mechanical power to the power available in the wind; tip speed ratio is the real measure of power of the wind a turbine can gain. This is presented in next equation (14), [7].:

$$Cp = \left[\left[(0.44 - 0.0167\beta) * \sin\left(\frac{\pi(\mu - 3}{15 - 0.3\beta}\right)\right] - \left[0.00184(\mu - 3)\beta\right]\right]$$

Tip speed ratio $\mu = \dfrac{\omega * Rr}{v}$ And the rotor speed:

$$\omega = \int_0^{1500} \left(\frac{\omega b}{2H}\right) * \left(Tm - Te - (\omega * F)\right) \qquad (15)$$

Where H is the equivalent inertia constant for both wind turbine and induction generator rotors and F is the friction factor [1],[5].

SEIG equations are presented by Matlab Simulink to form the equivalent circuits as follows respectively: stator voltage and current in direct axis, stator flux in quadrant axis, electrical / mechanical power, synchronous speed and load current as shown in Fig. 3. Also transmission line presentation for grid connection is shown in Fig.4.

3. Proportional Integral Controllers

The Proportional plus Integral (PI) controller is a conventional controller which has two gains to adapt the output of the controller. The first gain is called (KI) Integral gain, and the second one is called (KP) Proportional gain. It is applied on the system under study by using fixed gains. Both the integral gain and proportional gain are chosen with reference to terminal voltage error (eV) or mechanical power error (Pmv).

Where (KIv) is the integral gain of PI reactive controller and (KPv) is the proportional gain of PI reactive controller. But (KIF) is the integral gain of PI active controller and (KPF) is the proportional gain of PI active controller.

The voltage or mechanical power errors are used as an input variable to the PI controller, then the output is used to regulate the duty cycle of the switching capacitor bank in the reactive controller. The output of active controller is utilized to tune up the pitch angle of the wind turbine, which leads to adjust the system frequency [1].

The duty cycle and the pitch angel values are adjusted from the setting of the PI gains. The mathematical model of SEIG / WECS is simulated using MATLAP / SIMULINK software. A suddenly disturbance due to change for the nearest load is simulated

4. Artificial Neural Network "ANN"; the reactive controller:

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to produce final decisions for problems that are too complex to be noticed by either humans or other computer techniques.

A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze. This expert can then be used to provide projections given new situations of interest & answer "what if" questions.

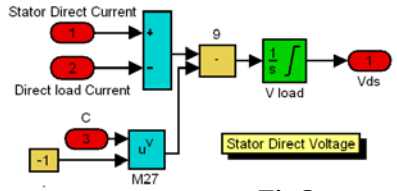


Fig.3.a

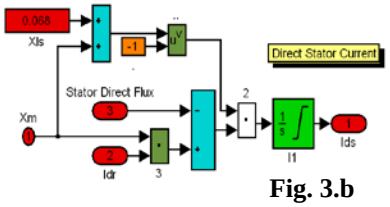


Fig. 3.b

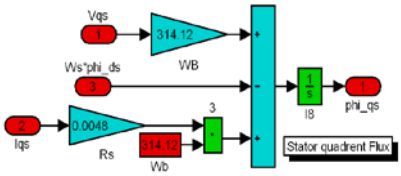


Fig. 3.c

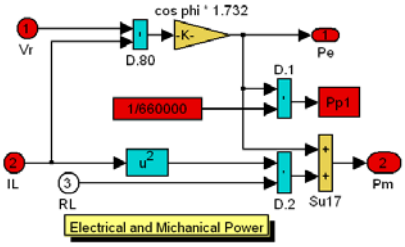


Fig.3.d

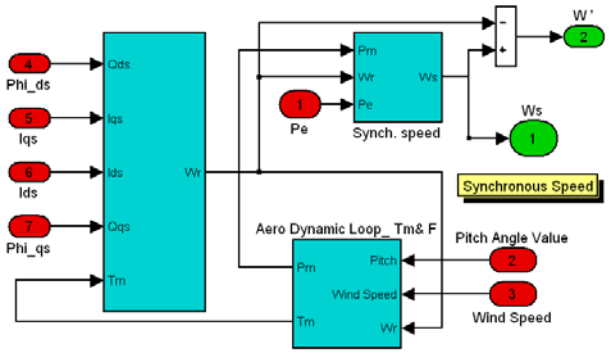


Fig.3.e

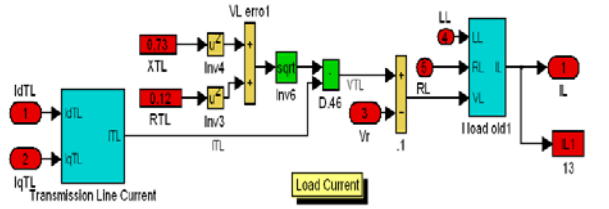


Fig.3.f

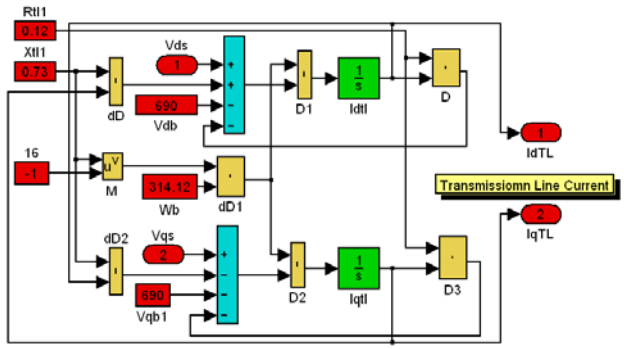


Fig.4:

This is done by ANN computations which are carried out in parallel, and special hardware devices are being designed and manufactured to take advantage of this capability.

The control scheme supposed to be used for controlling the operation of the wind turbine, uses "Feed forward propagation", where the main parameter which ANN monitors and reacts is the difference in voltage across generator terminals due to grid unbalance fluctuations.

This reaction is adapted by Back Propagation looping, where the voltage signals traveling in both directions of Input & Output neurons by introducing loops in the network.

Feedback is highly recommended to get perfect "trained" ANN output decision, which is considered to be very powerful and can get extremely complicated, by this design, the network will act dynamically; and the state is changing continuously until an equilibrium point is achieved, and will be remained until the input changes and a new equilibrium needs to be found.[2],[8],[9].

4.1 Design of ANN to control SEIG voltage stability

Induction machine connected to the grid operates as externally excited induction generator, while taking the excitation current from the grid, for which the operating speed of the machine must be greater than the synchronous speed. The output voltage and frequency should not change with loading conditions but active power generated by the machine is a function of slip. SEIG connected across the capacitor bank when driven by the wind within the limits of productive speeds. Under these conditions the capacitor bank meets the total reactive power requirements of the machine and load. Active power generated by the rotor is delivered to the load through stator.

The input of the ANN will be voltage error signal; the main goal of the control is to maintain this voltage signal steady.

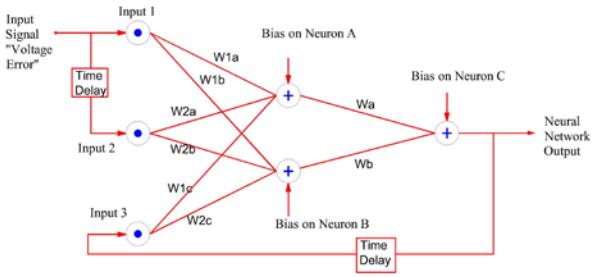


Fig. 5

This is done by applying the output of the network on the capacitor bank control to change its value accordingly [1],[10].

As illustrated in Fig. 5 the input layer of ANN design consists of three nodes, their inputs are firstly the initial data of the network which is the error in voltage signal, secondly the previous record of the error to be compared with the current value, and this is the main "learning target" for perfect adaptation of the output, lastly the previous record of the ANN output. The inputs then are subjected to a hidden layer, and then, weights of the networks modify the hidden layer intermediate results. Finally the output layer decides the best decision for the appropriate value corresponds to the correct capacitance of reactive compensator. Error calculation is done in all stages. That is, the neurons in hidden layer receives the adapted input signals from each node of the input layer, then it collects all and apply a "tansig" logistic function which was chosen as best try & error resulting function, then delivers the result after modifying the weights again plus error calculations to last stage. All these steps will be described mathematically as follows:

$$NNout = (Fa * Wa) + (Fb * Wb) + Bc \qquad (16)$$

Adaptation on weights affecting on the input signal of output layer neuron (Wa & Wb) are described as:

$$Wa = (Wai) + (Fa * \alpha r * De) + Bc \qquad (17)$$

$$Wb = (Wbi) + (Fb * \alpha r * De) + Bc \qquad (18)$$

The error calculation for output layer:

$$De = (Ve * NNout) * (1 - NNout) \quad (19)$$

Regarding the Input layer adaptation, the weights and error calculations for hidden layers are as follows:

$$Dec = (Wa * De * Fc) * (1 - Fc) \quad (20)$$

$$Ded = (Wa * De * Fd) * (1 - Fd) \quad (21)$$

$$Wc = (Wci) + (Ve * \alpha r * Dec) \quad (22)$$

$$Wd = (Wdi) + (Ve * \alpha r * Ded) \quad (23)$$

Where, Nx is an output of neuron x. Ve is the voltage error of applied voltage on generator terminals. Wxi is initial weight value applied on neuron x. Bx is the bios value applied on neuron x. NNout is the output value of the neural network. Doe, Dhe are error calculations for output and hidden layers respectively. Fx is logistic function applied on neuron x output. αr is the learning rate of the network.

The multilayer back propagation feed forward NN was used to develop a model that provides a good estimate of magnetization characteristics. [1].

In this network, Ve is used as input and Fc as output for training purpose. The network is set with 'tansig" activation function at the middle layer and 'purelin' activation function at the output layer. The design of the network and selection of optimum training parameters are performed by trial and error as per practical guide of on line data collected from running turbine. Table 1 shows an example of some practical online data given for some turbine parameters. A presentation of ANN model under test in shown by Matlab Simulink in Fig. 6.

Date/Time	Wind Speed (m/s)	Power (KW)	Pitch angle (Degree)
02.06.2008 11:20	9.5	476	-1.2
02.06.2008 11:30	9.9	517	-1.1
02.06.2008 11:50	9.7	503	-1.1
02.06.2008 12:00	12.2	660	5.7
02.06.2008 12:10	12.2	657	5.8
02.06.2008 12:20	11.9	653	4.2
02.06.2008 12:30	12.1	657	5.1
02.06.2008 12:40	12.0	656	4.9
02.06.2008 12:50	11.1	620	1.7
02.06.2008 13:00	12.9	660	7.9
02.06.2008 13:10	12.8	660	8.0

5. Results

Practically the conventional controllers of wind turbines, are very slow in responding to wind variations and load disturbances, and under failsafe strategy of this type of controller cause the wind turbine disconnect from grid for a long time "in minutes" until steady state of power and wind parameters are fulfilled then it reconnect the generator again, not only this but also in grid repeated drops in short period, a manual restart of the controller must be done. This is always causes low power production quality and various stop due to errors during normal operation, even in strong and productive wind speeds.

Applying ANN to the controller enhances the performance of the turbine's operation, this is done in the assumed controller where a sudden disturbance in the grid load and wind speed is conducted to outline the system characteristics' performance and to examine how this system can reaches steady state condition. A simulation for this dynamic case study system is conducted using MATLAB SIMULINK package.

In the system under study, the inputs to the controller are voltage, change of voltage (voltage error) and the output of the controller, these perform the duty cycle (λ) which is used to compute the effective capacitor bank value (C) as in equation (11). The duty cycle as a function of controller output is used as an input to semiconductor switches to change the capacitor bank value according to the need for the effective value of the excitation. Accordingly, in the semiconductor switching technique, the terminal voltage is controlled by adjusting the self-excitation through automatic switching for the capacitor bank.

Reactive control is required to overcome any disturbances in utility grid voltage during wind turbine operation.

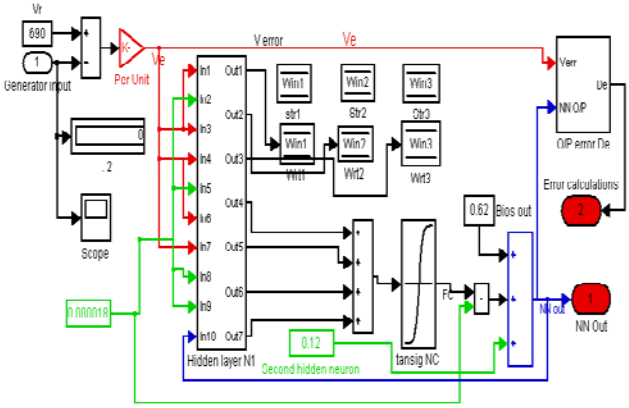


Fig.6

Table 1. Sample of blade pitch angle value corresponding to the wind turbine parameters at the same instance.

A capacitor bank unit is used in stages according to present VAR rating of the machine [1],[3],[5].

The control action can he carried out at any time of the power cycle so that it can provide a smooth variable source of reactive power. This leads to keep a constant terminal voltage equal to the rated value at different loads and different prime-mover speeds. The steady state results obtained are very much acceptable. Applications of an efficient on-line Artificial Neural Network (ANN) based power system stabilizer (PSS) for a wind turbine is

presented. The on-line training technique is used to update the weights and biases of each ANN controller, using the on-line back propagation (BP) algorithm. The ANN utilizes the speed deviation of the associated generator via the BP algorithm to enhance the overall dynamic response.

Fig. 7 illustrates the adaptation of applying ANN as an active control to the terminal voltage of the generator, where it help the system at starting of grid connection to reach the steady state faster and smoother than conventional controller. In addition, a sudden impact in voltage was fed to the scheme at time 20 and also derived the voltage to stability faster and smoother. In case of higher voltage drops, ANN can handle more fluctuations with connection to the electrical grid, where the damping period for the rising curve was reduced. This means increasing the turbine availability. But conventional controller In this case disconnects the generator completely from grid until voltage returns to rated values, which may lasts for multiply minutes.

Fig. 8 is another successful example for ANN adaptation but here for load current, a fast stability is obtained by ANN in both starting of generation process and in load impact, where Sudden load unbalance is applied on the system and ANN (light line) helps the load current reaches steady state faster than normal case (darker line).

On the other hand for active control, the ANN helps the system runs even when subjected to sudden wind speed fluctuation, as shown in Fig. 9 which shows per unit frequency behavior, the system was examined to sudden change in wind speed. Normally the turbine disconnects the generator from grid until a real value for wind speed is achieved, but with ANN a perfect adaptation for the rotor frequency is obtained to reserve the production process while wind speed values varies in large scales.

Finally and as a meter for a wind turbine performance a representation for the effect of ANN on the capacity factor is shown in Fig. 10 where the average value for Cp (green line) is higher than the same parameter in case of using conventional controller.

6. Conclusion

This research covers the control design and dynamic performance of induction generator driven by grid connected wind turbine. Molding and control design for WECS using self excited induction generator is presented, the scheme consists of three phase induction generator driven by horizontal axis wind turbine and interfaced to the electrical grid utility through an overhead transmission line. This utility grid is subjected to variances in load in addition to wind fluctuations, which is the case in Zafarana wind farm along the red sea. Artificial intelligence controller is applied to the scheme to enhance the availability and power production of the wind turbine.

Different control methods of varying degrees of complexity have been proposed and used for the scalar control of induction machine in isolated stand alone wind turbines, where the proposed neural network in this case determines the required magnitude and frequency of the injected rotor voltage to maintain a constant stator voltage magnitude and frequency.

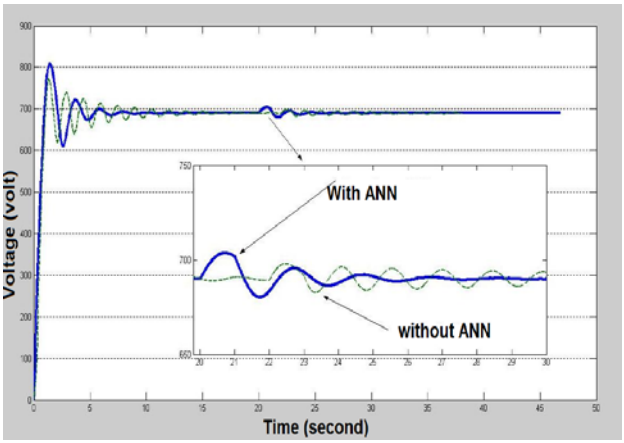


Fig.7

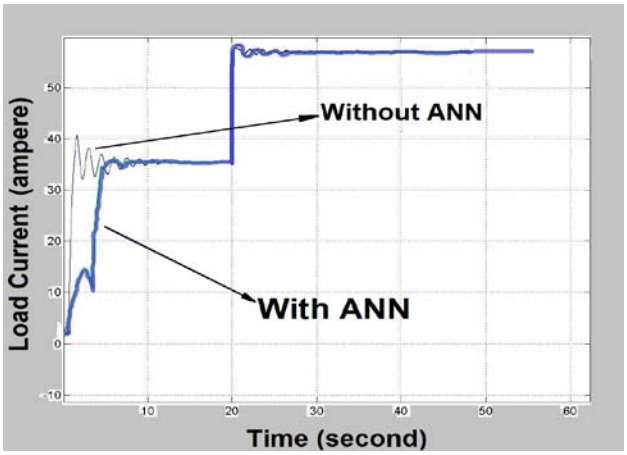


Fig. 8

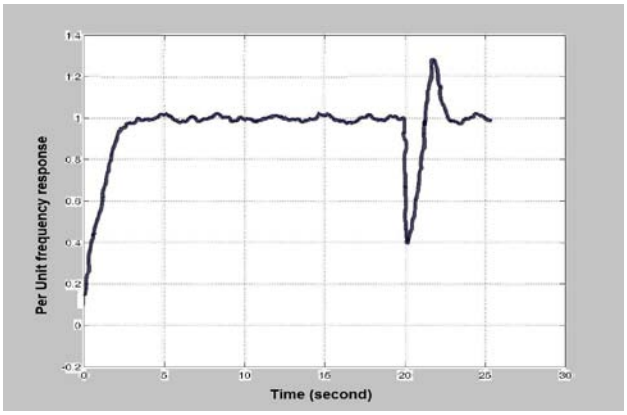


Fig. 9

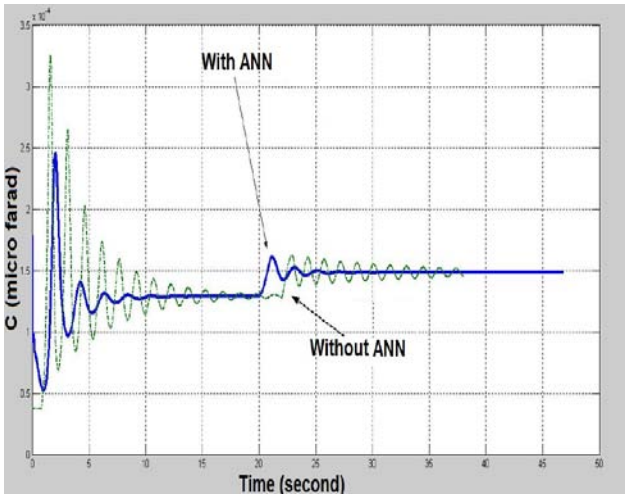


Fig.10

Provided that the main control strategy is based on regulating the machine flux level to be kept constant and closed to its nominal value as a variable speed wind turbine. The nature of application dictates the acceptance of particular method [10]. But the main difference in this paper is the direct connection to power utility grid, the matter which oblige the system to work under constant frequency, and then the strong impulses in load and/or voltage fluctuations is considered to be the main challenge for the controller to maintain the generated connected without failure.

The mathematical model of SEIG driven by WECS is simulated using MATLAB / SIMULINK package to solve its mathematical equations. Meanwhile, the neural network based controller has been developed
for the system under study. The controller acts as a reactive booster which is used to adjust and regulate the terminal voltage at the rated value, through the variation of the self-excitation using switching capacitor bank by controlling duty cycle. In addition, it acts as an active controller which regulates the input power to the generator, and thus maintains the stator frequency constant via changing the value of the blade pitch angle for the blade of the wind turbine.

The ANN utilizes the deviation in the terminal voltage to update the connection weights in different layers of the ANN through back propagation (BP) algorithm, where static compensator is used to drive the reactive controller for the machine to regulate the terminal voltage. In addition, the system under study is equipped with another ANN to regulate the mechanical power through the BP to adjust the blade angle of the WECS according to wind speed records, which leads the mechanical power input to be controlled using the blade pitch angel to represent "active control". This controller is used also to control the frequency of the overall system.

References:

[1] Sabry Mohamed Aly Mokymar, "Enhancement of The Performance of Wind Driven Induction Generators Using Artificial Intelligence Control"; Ain Shames University Faculty of Engineering - Dctoral thesis 2007

[2] Suresh H. Jangamshetti, "Dynamic Simulation Model of W.E. Conversion System", Senior Member, IEEE 2002.

[3] Andreas Petersson, "Analysis, Modeling and Control of Doubly-Fed Induction Generators for Wind turbines", 2005.Chalmers University of Technology Sweden 2005.

[4] M.Sasikumar and S.Chenthur Pandian, "Performance Characteristics of Self-Excited Induction Generator fed Current Source Inverter for Wind Energy Conversion Applications". International Journal of Computer and Electrical Engineering, Vol.2, No.6, 2010.

[5] K.Natarajan, S.Sivakumar, "Modeling & control design for wind energy power conversion using SEIG", IEEE transactions on Energy Vol EC-2 No. 3, 1987.

[6] K.S. Sandhu and S.P.Jain. "Steady State Operation of SEIG with Varying Wind Speeds". International Journal of Circuits, Systems and Signal Processing Issue 1, Volume 2, 2008.

[7] Ezzeldin S.Abdin & Wilson Xu, "Control design & dynamic performance analysis of a wind turbine induction generator unit, Senior member IEEE 2001.

[8] Raj Kumar Bansal , K. S. Sandhu and Ashok Kumar Goel, " Application of Artificial Neural Network for Analysis of Self-Excited Induction Generator", JCS&T Vol. 6 No. 2, 2006.

[9] Christos Stergiou & Dimitrios Siganos, " Neural Networks", Imperial College, Coputing Dep. issues, 1989

[10] L.Rajaji and C.Kumar."Neural network controller based induction generator for wind turbine applications" Indian Journal of Science and Technology Vol.2 No 2, 2009.

A Recurrent Neural Sleep-Stage Classifier Using Energy Features of EEG Signals

Jeen-Shing Wang[1], **Ya-Ting Yang**[2], **Chung-Yao Hsu**[3],**and Yu-Liang Hsu**[1]
[1]Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan, R.O.C.
[2]Institute of Education & Teacher Education Center, National Cheng Kung University, Tainan, Taiwan, R.O.C.
[3]Department of Neurology, Kaohsiung Medical University Hospital, Kaohsiung, Taiwan, R.O.C.

Abstract - *This paper presents a recurrent neural classifier to automatically classify sleep stages based on energy features of EEG signals by using only one single EEG channel (Fpz-Cz). The energy features are extracted from characteristic waves of EEG signals which can characterize different sleep stages individually. The recurrent neural classifier takes the energy features extracted on 30s epochs from EEG signals and assigns them to one of the five possible stages: Wakefulness, NREM 1, NREM 2, SWS, and REM. Eight sleep recordings obtained from Caucasian males and females without any medication are utilized to validate the proposed method. Moreover, the feedforward neural network and probabilistic neural network are also presented for comparison the performance of the proposed recurrent neural classifier with the same features. The classification rate of the recurrent neural classifier is better than that of the two neural classifiers. The results demonstrate that the proposed recurrent neural classifier with energy features of the characteristic waves of EEG signals can classify sleep stages more efficiently and accurately using only a single EEG channel.*

Keywords: Sleep stage, EEG, energy features, recurrent neural classifier

1 Introduction

Sleep disturbance causes some sleep disorders such as sleep apnea, insomnia, narcolepsy, etc. Nowadays, how to effectively diagnose and treat patients with sleep-related complaints has been heavily discussed within the sleep medicine community. In clinical diagnosis, the sleep stage classification of polysomnographic (PSG) measurements, which are composed of electroenphalogram (EEG), electromyogram (EMG), and electrooculogram (EOG), is an important and complex process since it involves diverse signal processing technologies and domain-experts' interpretation. PSG is generally divided into epochs of 20 or 30s, and then are classified into one of Rechtschaffen and Kales (R&K) rules by sleep technologists visually [1]. Based on the R&K rules, different sleep stages can be separated into the following six stages from sleep recordings in adults: wakefulness (WA), non-rapid eye-movement sleep stage 1 (NREM 1), NREM sleep stage 2 (NREM 2), NREM sleep stage 3 (NREM 3), NREM sleep stage 4 (NREM 4), and rapid eye-movement (REM). Sometimes, NREM 3 and NREM 4 are pooled into a single stage, slow wave sleep (SWS). In general, the sleep stages of PSG recordings are visually determined by doctors or physicians in clinical diagnosis and treatment. It is a time-consuming work and the agreement of individual experts is inconsistent. Studies have shown that even among expert coworkers, the agreement rate in sleep stage classification is less than 90% [5]. In this paper, we propose a sleep stage classification system that performs data acquisition, signal preprocessing, feature extraction, and classification in a fully automatic manner using a single EEG channel. First, the sleep EEG recordings are obtained from Sleep-EDF database [6]. In the signal preprocessing, the bandpass filters are utilized to acquire characteristic waves of EEG signals. Sequentially, the energy feature of each characteristic wave is extracted. Finally, we present the Elman recurrent neural classifier, which is capable of dealing with time series signals, to classify human sleep stages. In addition, we also compare the classification performance of the proposed recurrent neural classifier with two types of neural classifiers, a conventional feedforward neural network (FNN) and a probabilistic neural network (PNN). The advantage of our approach is that a sample set of energy features of characteristic waves of EEG signals with the proposed recurrent neural classifier can classify sleep stages efficiently using just a single EEG channel. The rest of this paper is organized as follows. In Section 2, the whole database, classification procedure, and a recurrent neural classifier for classifying sleep stages is described in detail. The results are presented in Section 3 and discussed in Section4. Finally, the conclusions are presented in Section 5.

2 Methods

The automatic sleep stage classification scheme includes data acquisition, signal preprocessing, feature extraction, and classification. The block diagram of the proposed scheme is shown in Fig. 1. In this paper, the sleep recordings used for sleep stage classification are collected by a single EEG channel. The sleep EEG recordings are obtained from Sleep-

EDF database. In the signal preprocessing, we have to use six bandpass filters to obtain six characteristic waves of EEG signals, respectively. Sequentially, we extract the energy features from the six characteristic waves. Finally, a recurrent neural classifier is utilized to recognize sleep stages of patients based on the extracted features.

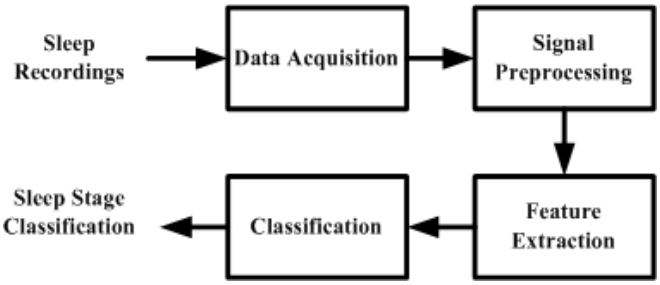


Fig. 1. Block diagram of the automatic sleep stage classification scheme.

2.1 Data Acquisition

The sleep recordings utilized in this study are obtained from the Sleep-EDF database which is available from the PhysioBank. The full recordings include eight sleep recordings obtained from Caucasian males and females without any medication. The eight subjects were aged from 21 to 35. For the first four sleep recordings (sc4002e0, sc4012e0, sc4102e0, and sc4112e0), the subjects were ambulatory healthy volunteers, and the recordings were obtained during 24 hours in their normal daily life in 1989. For the other four sleep recordings (st7022j0, st7052j0, st7121j0, and st7132j0), the subjects had mild difficulty falling asleep but were otherwise healthy, and the recordings were obtained during a night in the hospital in 1994. The sleep recordings include horizontal EOG, and two EEG channels (F_{pz}-C_z and P_z-O_z). All signals were digitized at a 100 Hz sampling rate. In this paper, we only used a single EEG channel (F_{pz}-C_z channel) to perform automatic sleep stage classification tasks. EOG information and the other EEG channel (P_z-O_z channel) in the sleep recordings were ignored. The EEG data was divided into 30s epochs for off-line analysis. The EEG data used in this study consists of 2880 randomly selected epochs. The training subset was composed of 1920 epochs (240 epochs adopted from each subject) which were chosen to train the neural classifiers. In addition, the testing subset was composed of 960 epochs (120 epochs adopted from each subject) which were used to validate the classification accuracy. According to the R&K rules, each EEG epoch can be classified as one of the following five different sleep stages: WA, NREM 1, NREM 2, NREM 3 & 4 (or SWS), and REM. Not every epoch can be 100% fit in a specific stage. The decision depends on which stage properties are present the most and that is sometimes difficult to discriminate.

2.2 Signal Preprocessing

Sleep recordings are composed of WA, NREM 1, NREM 2, NREM3 & 4, and REM sleep stages. Normally, EEG signals are composed of alpha (α), beta (β), theta (θ), delta (δ), spindle, sawtooth, and *K* complex waves. In this paper, the definitions of the boundaries between the frequency bands are defined as: alpha = [8-13 Hz], beta = [12-30 Hz], theta = [4-8 Hz], delta = [0.5-2 Hz], spindle = [12-14 Hz], sawtooth wave = [2-6 Hz], and *K* complex = [1 Hz]. In different stages, EEG signals can be characterized by different features or waves. The characteristic waves of EEG signals for the abovementioned stages are shown in Table I. From Table I, we can find apparently that spindle wave, sawtooth wave, and delta wave are dominant in NREM 2, REM, and NREM 3 & 4, respectively. In addition, alpha and beta waves are dominant in WA, while theta wave is dominant in NREM 1. Therefore, the discriminative information of EEG signals can be carried by different kinds of specific frequency components. Hence, the philosophy of our signal preprocessing method is to utilize six finite impulse response (FIR) bandpass filters in 8-13 Hz, 12-30 Hz, 4-8 Hz, 0.5-2 Hz, 12-14 Hz, 2-6 Hz to separate alpha, beta, theta, delta, spindle, and sawtooth waves of the EEG signals, respectively. They were calculated using fast Fourier transformation (FFT) and the best results were obtained by using an FFT-filter with Hamming window.

Table I
CHARACTERISTIC WAVES OF EEG SIGNALS FOR EACH SLEEP STAGE

Sleep Stage	Characteristic Wave
WA	Alpha (8-13 Hz), Beta (12-30 Hz)
NREM 1	Theta (4-8 Hz)
NERM 2	Spindle (12-14 Hz), *K* complex (1 Hz)
NREM 3 & 4	Delta (0.5-2 Hz)
REM	Alpha (8-13 Hz), Beta (12-30 Hz), Theta (4-8 Hz), Sawtooth wave (2-6 Hz)

2.3 Feature Extraction

To obtain automatic sleep stage neural classifiers, two problems are considered: 1) to extract adequate input features from EEG signals for neural classifiers; and 2) to choose suitable neural classifiers that can obtain the best results on sleep stage classification problems. Feature extraction is an important step and usually influences the classification performance for any sleep stage classification system. Once we obtain the above six characteristic waves from the EEG signals, we can extract the energy features from the six waves. The energy of each wave of each 30s epoch is the summation of the magnitudes of squared components of the signal in an epoch defined as below:

$$\text{Energy} = \sum_{i=1}^{N} X_i^2, \qquad (1)$$

where X_i is the i^{th} component of the epoch and N is the total number of data in an epoch. Thus, each epoch can be parameterized by the following six features: $E_{(EEG, alpha)}$, $E_{(EEG, beta)}$, $E_{(EEG, theta)}$, $E_{(EEG, delta)}$, $E_{(EEG, spindle)}$, and $E_{(EEG, sawtooth)}$. Once we obtain the energy features as inputs for neural classifiers, we must choose suitable neural classifiers to obtain the best classification accuracy.

2.4 Neural Classifiers

Among various intelligent-based approaches, neural networks have been recognized as one of the most attractive classifiers for categorizing human sleep stage problems because of their learning capability and robustness. Besides, fault tolerance is another good property of neural networks and it is important in sleep research experimentation where recording phases can be partially impeded and the processed data can be blurred due to undesirable events. Generally, neural networks can be classified into two categories: FNNs and RNNs. In FNNs, the signal flows are all connected in only one direction from input to output of the networks. On the contrary, RNNs contain not only feedforward connections but also feedback connections. In this study, we utilize a notable Elman RNN as a recurrent neural classifier for categorizing human sleep stages [3]. For classification problems, RNNs can make efficient use of temporal information in the input sequence. In this paper, we use the dynamic behavior of the RNN to categorize input features into different specified sleep stages. In addition, we also compare the classification performance of the proposed RNN with two types of neural networks, a conventional FNN and a PNN. The reason for comparing the classification performances is that FNNs have been widely used for classification problems of biomedical signals effectively and PNNs are mainly designed for dealing with classification problems. The comparison and discussion about the abovementioned neural classifiers are provided in following section. The Elman network consists of an input layer, a hidden layer, a context layer, and an output layer. In this paper, $\mathbf{u} = [u_1, u_2, ..., u_p]^T$ and $\mathbf{y} = [y_1, y_2, ..., y_m]^T$ are the input and output vectors, respectively, where p represents the number of elements in the input feature set and m is the number of classes. The input vector of the Elman network is composed of the six extracted energy features (i.e. $p = 6$), while output vector of the Elman network consists of the five sleep stages (i.e. $m = 5$). Note that the i^{th} element in the target output vector is "1" and the others are "0" as the corresponding epoch belongs to the i^{th} class. When testing the trained network, all elements of the output vector of the trained network will compete against each other. Then the winner element will be set to "1" and the lost elements to "0".

3 Results

To validate the classification performance of the proposed automatic sleep stage classification scheme with the recurrent neural classifier, the classification accuracy is presented in this section. The sleep recordings, which include 2880 epochs (1920 epochs for training set and 960 epochs for testing set) of 8 different individuals, were used to demonstrate the effectiveness of the proposed classification scheme and the recurrent neural classifier. Note that only the Fpz-Cz EEG channel was used to obtain the results of the classification accuracy which are presented below. The classification accuracy was obtained using the six energy features ($E_{(EEG, alpha)}$, $E_{(EEG, beta)}$, $E_{(EEG, theta)}$, $E_{(EEG, delta)}$, $E_{(EEG, spindle)}$, and $E_{(EEG, sawtooth)}$) of each characteristic wave of each EEG epoch. The results are distributed into two parts which focus on 1) the general performances of the Elman recurrent neural classifier, FNN classifier, and PNN classifier for classifying sleep recordings and 2) the classification accuracy of the individual stages.

3.1 General Performance of the Neural Classifiers

To obtain parsimonious network structures for the proposed neural classifiers, we created 11 proposed Elman networks and FNNs with different sizes (from 1 to 10 and 15), respectively. Note that the structure size of the PNN was based on the training samples. Figs. 2 and 3 illustrate the comparison of the classification performance of different numbers of the hidden neurons of the Elman networks and FNNs, respectively. From Fig. 2, we found that neurons of the Elman network that rank after 7 did not make noticeable contributions to the overall classification performance. From Fig. 3, we could also find that the performances of the classification scheme with extra neurons were pretty similar to the performance of the scheme with six hidden neurons of the FNNs. Once the structure sizes of the neural classifiers were decided, the experiments with the neural classifiers were repeated 10 times for different initial weights respectively, and the results were average. The classification performances with the neural classifiers for individual subjects are shown in Table II. For the Elman recurrent neural classifier, the average classification accuracy was 80.9%-91.8% for different subjects. For the FNN classifier, the average classification accuracy was 70.6%-87.3% for different subjects. For the PNN classifier, the average classification accuracy was 70.8%-86.7% for different subjects. The overall average classification accuracy of the Elman recurrent classifier, FNN classifier, and PNN classifier were 87.2%, 81.1%, and 81.8%, respectively. In addition, we selected subject "sc4112e0" to highlight the performances of the proposed scheme using the neural classifiers. Fig. 4 shows the best results of sleep stage classification, which were obtained by manual scoring and the proposed neural classifiers, respectively. Obviously, the four hypnograms were quite similar. The best classification performances for the subject using the Elman network, FNN, and PNN were 93.3%, 89.2%, and 86.7%, respectively. Obviously, the three neural classifiers with the proposed classification scheme could all achieve satisfactory accuracy only for the Fpz-Cz EEG channel. Moreover, the classification results showed that the Elman network was the

best neural classifier for the sleep stage classification problems.

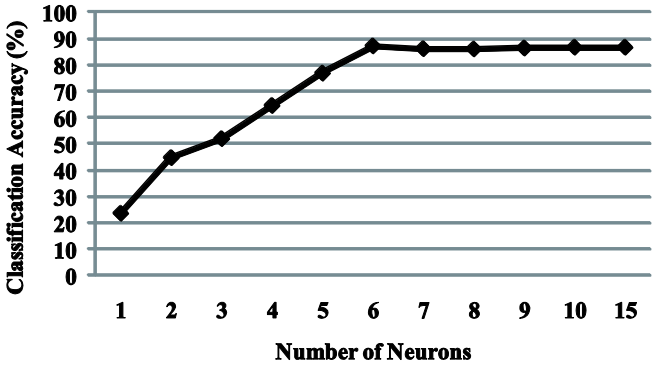


Fig. 2. Classification accuracy using different size of hidden layer of the Elman recurrent neural classifier.

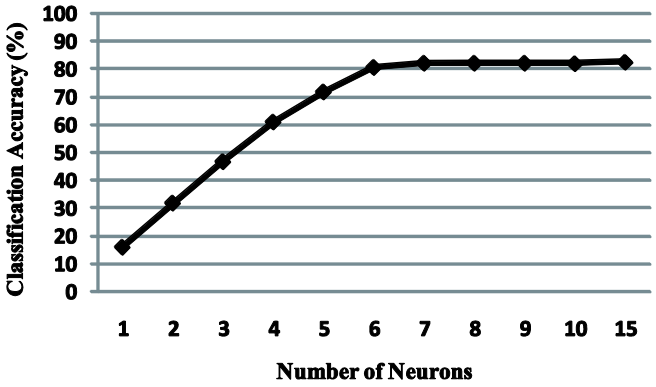


Fig. 3. Classification accuracy using different size of hidden layer of the FNN classifier.

Table II
CLASSIFICATION ACCURACY OF THE NEURAL CLASSIFIERS FOR INDIVIDUAL SUBJECTS

%	Elman Network	FNN	PNN
sc4002e0	87.7 ± 1.5	82.6 ± 2.0	70.8
sc4012e0	80.9 ± 0.8	70.6 ± 1.8	81.7
sc4102e0	87.1 ± 1.5	82.3 ± 1.4	81.7
sc4112e0	91.8 ± 1.4	86.3 ± 1.6	86.7
st7022j0	85.7 ± 0.7	80.1 ± 1.5	75.8
st7052j0	89.4 ± 2.0	87.3 ± 3.2	83.3
st7121j0	86.2 ± 1.4	72.8 ± 2.9	83.3
st7132j0	88.7 ± 2.6	84.2 ± 2.2	85.0
Average	87.2	81.1	81.8

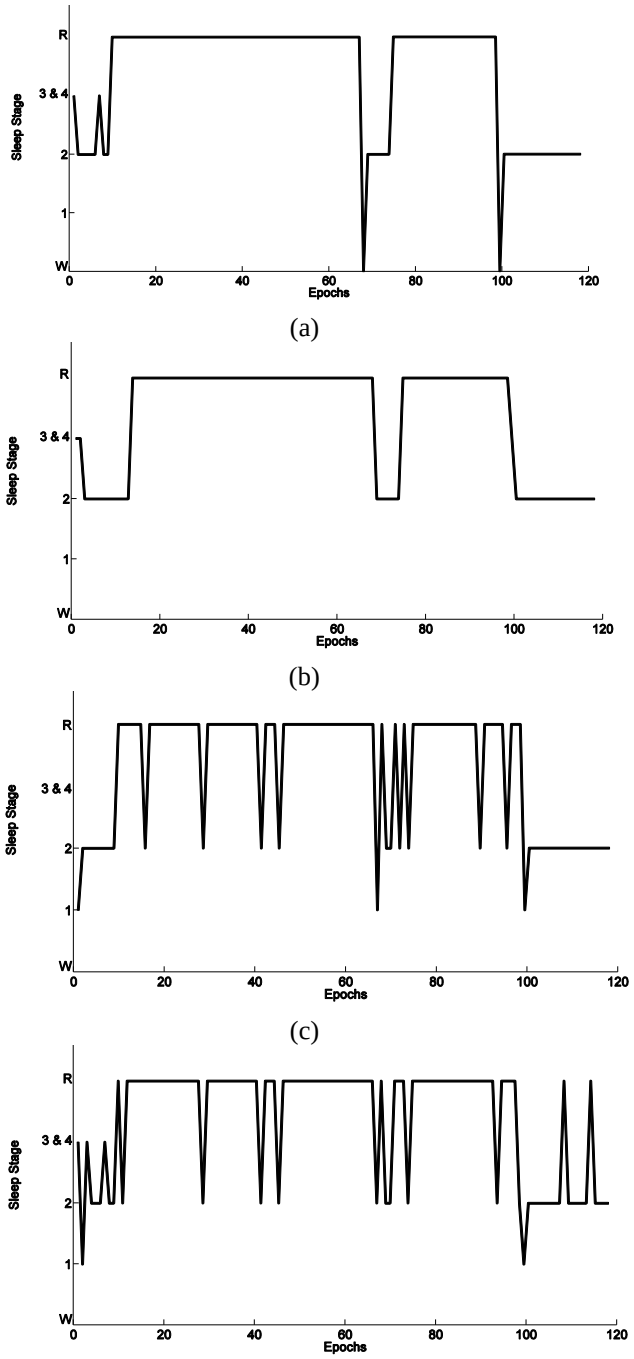


Fig. 4. Results of classification of sleep stages for "sc4112e0" obtained by (a) visual classification; (b) Elman recurrent neural classifier; (c) FNN classifier; and (d) PNN classifier.

3.2 Classification Accuracy of the Individual Stages

The classification accuracy for the sleep stage classification using the Elman network, FNN, and PNN are shown in Figs 5, 6, and 7, respectively. Fig. 5 shows the classification accuracy of each individual sleep stage using the

Elman network. The poorest performance was the accuracy rate of 40% for NREM 1, while performances for the other stages were significantly better (70.8%-97.3%). Fig. 6 represents the classification accuracy of each individual sleep stage using the FNN classifier. The poorest performance was the accuracy rate of 30% for NREM 1, while performances for the other stages were significantly better (75.0%-86.2%). The classification performances of each individual sleep stage by using the PNN classifier are shown in Fig. 7. The poorest performances were the accuracy rate of 25.5% for NREM 1 and 39.6% for WA. The performances of other stages were significantly better (82.8%-90.3%). Obviously, the classification performance of NREM 1 was the poorest despite neural classifiers since almost all misclassified epochs were within the adjacent stage.

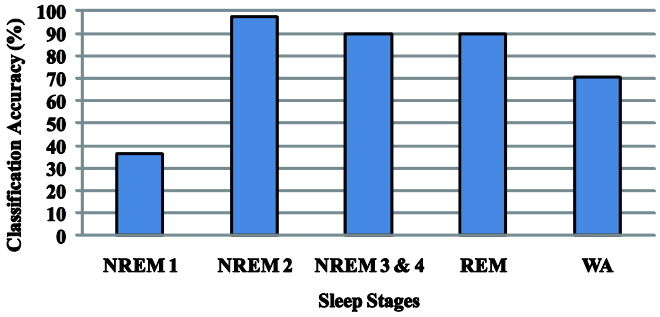


Fig. 5. Classification accuracy for each sleep stage using the Elman recurrent neural classifier.

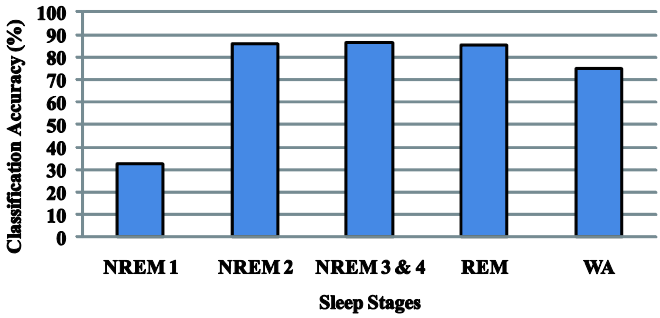


Fig. 6. Classification accuracy for each sleep stage using the FNN classifier.

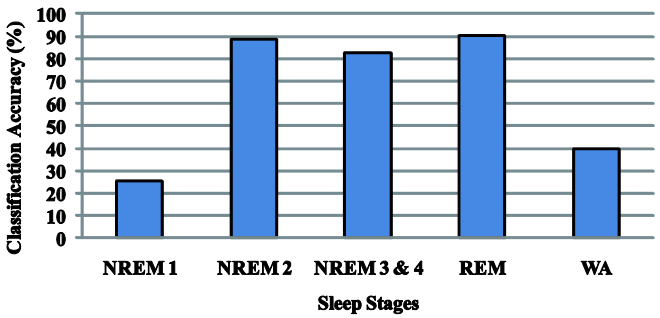


Fig. 7. Classification accuracy for each sleep stage using the PNN classifier.

4 Discussion

This study showed that neural classifiers could classify sleep stages using only a single EEG channel with satisfactory accuracy. Furthermore, the proposed recurrent neural classifier could obtain the best classification performance since its robustness to deal with EEG time series. The classification scheme categorized 30s EEG epochs into one of the five categories: WA, NREM 1, NREM 2, NREM3 & 4, and REM. After the data acquisition and signal preprocessing, the proposed feature extraction method was used to extract individual energy feature of the individual characteristic wave from the EEG epochs. After the feature extraction, the Elman network was utilized as a recurrent neural classifier for the sleep stage classification problems. In addition, we also utilized the FNN and PNN classifiers to validate that 1) the proposed three neural classifiers with the proposed classification scheme can all have satisfactory classification accuracy for dealing with a single EEG channel; and 2) the Elman recurrent neural classifier can obtain the best results over other neural classifiers.

For neural classifiers, if the proper features are selected, the classification performances can achieve satisfactory accuracy. The only a priori information used by the abovementioned neural classifiers was the amount of energy features associated with the characteristic waves of each stage and the number of output classes. Table II shows that the best classification accuracy using the abovementioned neural classifiers, respectively. The average classification accuracy of the Elman network, FNN, and PNN could achieve about 87.2%, 81.1%, and 81.8%, respectively. Obviously, the accuracy of the FNN and PNN was similar and the Elman network is the best classifier. The classification accuracy of the Elman network was 2.1%-13.4% higher than the accuracy of the FNN for different subjects. The classification accuracy of the Elman network was 2.9%-16.9% higher than the accuracy of the PNN for different subjects, except subject "sc4012e0". The Elman network is best capable of dealing with EEG time series. Hence, the Elman network has the capacity for handling many biomedical time series signals and classification tasks.

The classification accuracy of each stage using the abovementioned neural classifiers for all subjects is shown in Figs. 5, 6, and 7. The greatest inconsistency occurred in NREM 1, which was to be expected. The two major reasons for this situation using the neural classifiers are: 1) experts are usually inconsistent in the classification of sleep stage NREM 1 since it is a transition from wakefulness to the other sleep stages; and 2) NREM 1 is only between 3% and 6% of the night. However, in the neural classifier point of view, the number of training data can influence the classification performance, which means that a small training data has poorer classification accuracy than a large one. Hence, the classification performances of NREM 1 using neural classifiers were unsatisfactory. In addition, it is difficult to discriminate NREM 1 from REM stage using only EEG data. The presence of rapid eyes movement is the dominant feature

of REM stage. However, this feature was recorded in EOG signals, while it appeared as artifacts in EEG signals. From the results, most epochs of NREM 1 were misclassified as REM stage. Hence, how to accurately discriminate NREM 1 from REM stage using only a single EEG channel is a tremendous challenge in automatic sleep stage classification. From the results, the performances of NREM 1 and REM stages were acceptable using only a single EEG channel in this paper.

On the other hand, the classification performances of NREM 2, SWS, and REM sleep stages were at least about 83%. These errors were due to periods of transitions from one sleep stage to another. The reason for NREM 2 having good classification performance despite the neural classifiers is that NREM 2 takes about 50% of the night. From our literature review, some studies on automatic sleep stage classification showed that SWS could decrease the effect of transitions between NREM 3 and NREM 4 [2], [4], [7]. When these two sleep stages are pooled as SWS, the overall classification accuracy can be improved drastically. In this paper, we combined NREM 3 and NREM 4 into SWS stage and the agreement percentage of SWS was about 80-90%. In addition, most REM epochs were misclassified as NREM 2 because the REM stage could possibly look exactly like NREM 2 without spindles. The classification accuracy of the WA stage was about 40%-70% which was poorer than the above three sleep stages. The reason for the situation is that the number of WA epochs used in this paper was smaller than the other sleep stages.

5 Conclusions

The new automatic sleep stage classification scheme, applying the relative energy features of EEG signals and the recurrent neural classifier, has been proposed in this paper. The energy of the characteristic waves of EEG signals has been extracted as the input features for the recurrent neural classifier. From the results, it can be concluded that the extracted features, the relative energy of the individual characteristic waves of each stage, are the discriminating features for classifying sleep stages. Furthermore, the recurrent neural classifiers make these features highly suitable for sleep stage classification using a single EEG channel.

6 References

[1] A. Rechtschaffen and A. Kales, "A manual of standardized terminology, techniques, and scoring system for sleep stages of human subjects," in *UCLA, Brain Research Institute/Brain Information service*, Los Angeles, CA, 1968.

[2] J. M. Gaillard and R. Tissot, "Principles of automatic analysis of sleep records with a hybrid system," *Computers and Biomedical Research*, vol. 6, pp. 1-13, 1973.

[3] N. F. Güler, E. D. Ü beyli, and İ. Güler, "Recurrent neural networks employing Lyapunov exponents for EEG signals classification," *Expert Systems with Applications*, vol. 29, no. 3, pp. 506-514, 2005.

[4] J. Hasan, "Past and future of computer assisted sleep analysis and drowsiness assessment," *J. Clin. Neurophysiol.*, vol. 13, pp. 295-313, 1996.

[5] R. G. Norman, I. Pal, C. Stewart, J. A. Walsleben, and D. M. Rapoport, "Interobserver agreement among sleep scorers from different centers in a large dataset," *Sleep*, vol. 23, no. 7, pp. 901-908, 2000.

[6] B. Kemp, The Sleep-EDF Database, http://www.physionet.org/physiobank/database/sleep-edf/, 2006.

[7] E. Stanus, B. Lacroix, M. Kerkhofs, and J. Mendlewicz, "Automated sleep scoring: A comparative reliability study of algorithms," *Electroencephalogr. Clin. Neurophysiol.*, vol. 66, pp. 448-456, 1987.

A Linear Classifier Outperforms UCT in 9x9 Go

N. Sylvester, B. Lohre, S. Dodson, and P. Drake
Department of Mathematical Sciences
Lewis & Clark College, Portland, Oregon, USA
{nsylvester, blohre, sdodson, drake}@lclark.edu

Abstract - *The dominant paradigm in computer Go is Monte-Carlo Tree Search (MCTS). This technique chooses a move by playing a series of simulated games, building a search tree along the way. After many simulated games, the most promising move is played. This paper proposes replacing the search tree with a neural network. Where previous neural network Go research has used the state of the board as input, our network uses the last two moves. In experiments exploring the effects of various parameters, our network outperforms a generic MCTS player that uses the Upper Confidence bounds applied to Trees (UCT) algorithm. A simple linear classifier performs even better.*

Keywords: board game, Go, machine learning, Monte-Carlo methods, neural network

1 Introduction

Go is a board game that originated in China several thousand years ago [2]. Writing a program that plays Go well is difficult because of the large search space and the absence of powerful evaluation functions. Today the best human players are still able to beat the best computer players [7].

This paper presents a new, promising technique based on neural networks in a Monte-Carlo context. Section 2 describes Monte-Carlo techniques, neural networks, and previous work. Our neural Monte-Carlo technique is described in section 3. Experimental results are then presented, followed by conclusions and future work.

2 Previous Work

2.1 Monte-Carlo techniques

Monte-Carlo techniques use random sampling to solve complex problems [10]. In Monte-Carlo Go, this random sampling consists of playing a series of simulated games (playouts). The moves within each playout are chosen according to some policy. This policy often involves two parts: a primary policy that generates the early moves within each playout and a secondary policy that generates the remaining moves. The primary policy might be a search tree (as in Monte-Carlo tree search) or a neural network (as in this paper). The secondary policy is largely random, but may be heuristically biased to favor better moves. After each playout, the winner is determined; the policy is adjusted to encourage the winner's moves and discourage the loser's moves. Each playout is thus affected by the results of previous playouts. When time runs out, the program plays the move that fared best, e.g., the move that started the most playouts.

Monte-Carlo Tree Search (MCTS) is the dominant paradigm in computer Go [8]. MCTS uses a search tree as its primary policy. At each node in the tree, a move is chosen based on how many times it has been played and how many of those playouts resulted in wins. At the end of the playout, the number of plays and, if appropriate, the number of wins is incremented at each node encountered during the playout. A leaf is added to the tree representing the first move beyond the tree. The tree thus expands down the most promising branches.

One challenge in using this type of search is striking a balance between exploiting moves that show promise and exploring undersampled moves. A key breakthrough in MCTS was the invention of the Upper Confidence bounds applied to Trees (UCT) policy that occasionally visits such undersampled moves [6].

2.2 Neural networks

Artificial neural networks are a popular technique for supervised machine learning. A network is made up of computational units arranged in a weighted, directed, acyclic graph. Activation flows from the network input, potentially to "hidden" units, and on to the output units. Each unit's activation is a function of the weighted sum of the activations of previous units. This function may be the identity function, as in a linear classifier [10], or the logistic sigmoid function $1/\left(1+e^{-x}\right)$, as in a typical backpropagation network [9]. A network can be trained by gradient descent, adjusting the weights to reduce the difference between the desired and actual output of the network.

Past neural network Go research has focused on predicting moves of amateur and professional players rather than playing [11]. Others have trained their networks offline [3]. Our program joins a neural network and Monte-Carlo methods to play Go,

learning during the game. Since recent moves have been found to be a useful predictor of which move to make next [1], we use the previous two moves as the input to our neural network.

3 Neural Monte-Carlo

This paper proposes an artificial neural network [9, 10] as the primary policy. The intent is to improve generalization. In pure MCTS, no information is shared between nodes of the tree. Even if a good response B is found to a given move A, the response must be rediscovered from scratch if other, irrelevant moves are played before A. MCTS can be improved by sharing information between branches of the tree [1, 4]. We reasoned that a single network (rather than separate statistics maintained for each node) would allow useful sharing of information, while still responding to different situations appropriately. Furthermore, it might reduce the memory requirements of the program; if the network could be kept in cache, this might even increase speed.

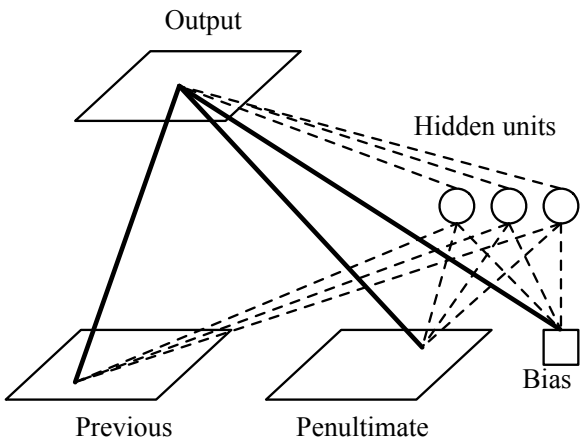


Figure 1: Network structure

Our neural network has an output unit for each point on the board; the legal move with the highest output is the network's choice. The inputs consist of one array of units indicating the previous move, another array indicating the penultimate move, and a single bias unit. There is an optional single layer of hidden units. In addition to the usual input-to-hidden and hidden-to-output connections, we have direct input-to-output connections (Figure 1). This avoids a bottleneck that results from trying to pass too much information though the hidden units.

The use of recent history, rather than a global view of the board, allows us to save a great deal of computation when determining the network output. Since only two input units (plus the bias unit) are activated at any given time, almost all of the weights can be ignored in any given update of the network.

Our player uses, as its primary policy, two of these networks (one for each player). While it would be possible to use the network as the entire policy, we found that it was more effective to only use the network for a fixed number of moves (see subsection 4.5).

4 Experimental Results

4.1 Methods

All of the experiments were run using Orego version 7.09. Orego options that would not interfere with the fairness of tests were left intact. Specifically, the escape-pattern-capture secondary policy suggested in [5] was used. Moves were only considered if they were on the 3rd or 4th line or were within a large knight's move of an existing stone. MCTS was represented by the Orego MctsPlayer (which uses UCT). As the network did not have any heuristic bias or pre-training, the MCTS priors (virtual wins granted to heuristically good moves) were set to 0. The transposition table was left intact in MCTS, but RAVE and LGRF-2 were not used.

Experiments were run on a CentOS Linux cluster of five nodes, each with 8 GB RAM and two 6-core AMD Opteron 2427 processors running at 2.2GHz, giving the cluster a total of 60 cores. Experiments were run using a single thread. The program was run with Java 1.6.0, using the command-line options -ea (enabling assertions) and -Xmx1024M (allocating extra memory). In all experiments, each condition involved 600 9x9 games (300 with Orego as black, 300 as white) using Chinese rules, positional superko, and 7.5 komi. Win rates are against GNU Go 3.8 running at its default level with the capture-all-dead option turned on.

4.2 Experiment 1: Hidden Units

The first experiment examined the effect of the number of hidden units on the strength of the player. We built networks with 0, 10, 20, and 40 hidden units and compared them to plain MCTS. The network weights were initialized randomly, uniformly distributed over the range [-0.5, 0.5). The network's learning rate was set to 0.5. The cutoff (the number of moves generated by the network in each playout before deferring to the secondary policy) was set to 10. The results are shown in Figure 2.

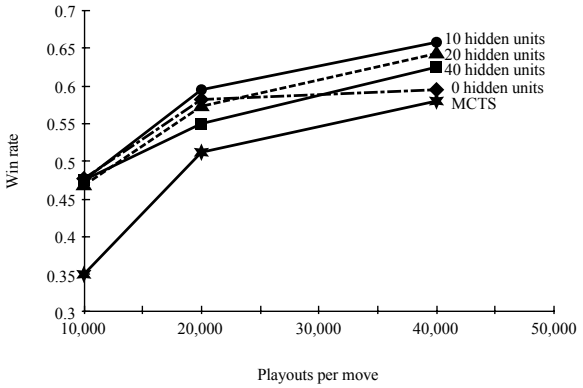


Figure 2: Win rate vs GNU GO as a function of number of hidden units

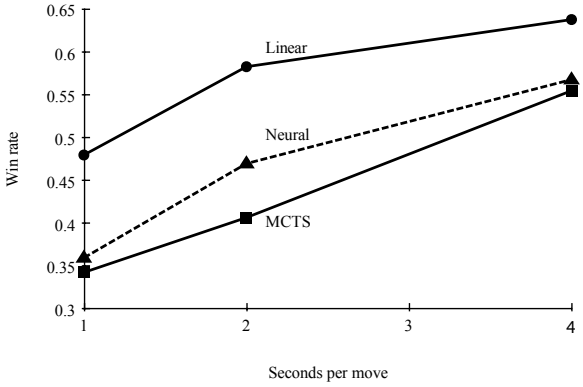


Figure 3: Win rate vs GNU GO for different players

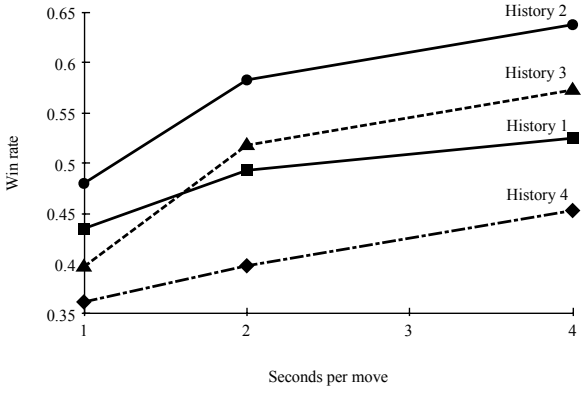


Figure 4: Win rate vs GNU Go as a function of history

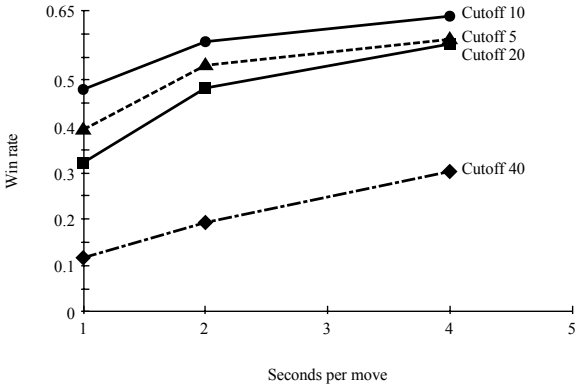


Figure 5: Win rate vs GNU Go as a function of cutoff depth

The network player significantly outperformed MCTS for all numbers of hidden units at 10,000 playouts per move; for 0, 10, and 20 hidden units at 20,000 playouts; and for 10 and 20 hidden units at 40,000 playouts ($p < 0.05$, two-tailed z-test).

The 10-hidden-unit network was significantly better than the 0-hidden-unit network at 40,000 playouts. In no other case did the number of hidden units make a significant difference. Specifically, at 10,000 playouts, the number of hidden units appears to make no difference at all.

While the addition of hidden units shows promise for larger number of playouts, we were concerned that any improvement in strength might be outweighed by the extra computation time. If hidden units are not used, we reasoned, we could save even more time by using simple linear units rather than sigmoid units. This is the subject of the next experiment.

4.3 Experiment 2: Linear Units

The second experiment explored how the time given per move affected the performance of different of players. Fixing the time, rather than the number of playouts, accounts more fairly for how efficiently different players use their time. We gave each player either 1, 2, or 4 seconds to make a move. We compared the plain MCTS; a neural player with learning rate 0.5, cutoff 10 and 0 hidden units; and a linear player with learning rate 0.01 and cutoff of 10. The linear network's weights were initialized to 0. The results are shown in Figure 3.

The linear player performed significantly better than the other two players in all three conditions. The neural player significantly outperformed MCTS at 2 seconds per move.

4.4 Experiment 3: History

We next examined how many previous moves should be used as input to the network. The third experiment varies this history parameter, using a linear player with learning rate 0.01 and cutoff 10. The results are shown in Figure 4.

There is a significant difference in the performance of a linear player with a history of 2 versus all other history levels at 2 and 4 seconds per move. The linear player with a history of 4 turned out to be significantly worse than all of the other players at 2 and 4 seconds per move.

These data show that looking farther back in history does not necessarily increase performance as one might expect. It may be that the distant past (more than two moves) is simply not well-correlated with what a player should do next.

4.5 Experiment 4: Cutoff

Our fourth experiment examined how many moves the network should make before deferring to the secondary policy. This cutoff parameter was varied in a linear player with history 2 and learning rate 0.01. The results are shown in Figure 5.

The cutoff value of 10 was significantly better than the other tested values at 1 move per second and was significantly better than cutoffs of 20 and 40 at both 2 and 4 moves per second.

This shows that including a secondary policy is important. This result contradicts the intuition that generating as many moves as possible from the learning network would be best. We hypothesize that preserving diversity in the playouts is more important than playing "good" moves.

5 Conclusions and Future Work

Our neural network performs significantly better than plain MCTS. A simple linear classifier is stronger still. In future work, we hope to improve our technique so that it is competitive with cutting-edge MCTS variations such as RAVE [4] and LGRF-2 [1] in 19x19 Go. Since prior knowledge helps advanced MCTS, it could be valuable to pre-initialize the weights in our network based on self-play or expert human games. Parallelizing our program should boost performance. We would also like to explore other inputs to the network.

6 Acknowledgments

This research was funded by the John S. Rogers Science Research Program at Lewis & Clark College and the James F. and Marion L. Miller Foundation.

References

[1] H. Baier and P. Drake. "The Power of Forgetting: Improving the Last-Good-Reply Policy in Monte-Carlo Go," in *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 2, no. 4, pp. 303-309, 2010.
[2] K. Baker. "The Way to Go." Internet: http://www.usgo.org/usa/waytogo/index.asp, accessed June 5, 2011.
[3] M. Enzenberger. "Evaluation in Go by a Neural Network Using Soft Segmentation," in *Proceedings of the 10th Advances in Computer Games Conference*, 2003, pp. 97-109.
[4] S. Gelly and D. Silver. "Combining online and offline knowledge in UCT," in *Proceedings of the 24th International Conference on Machine Learning*, pp. 273-280. 2007.
[5] S. Gelly et al. "Modification of UCT with Patterns in Monte-Carlo Go," INRIA, France, 2006. pp. 1-21.
[6] L. Kocsis and C. Szepesvári. "Bandit Based Monte-Carlo Planning," in *Machine Learning: ECML 2006*, 1st ed., vol 4214. J. Fürnkranz, T. Scheffer, and M. Spiliopoulou, Ed. Berlin: Springer. 2006, pp.282-293.
[7] M. Müller. "Computer Go," in *Artificial Intelligence*, 2002, pp. 145-179.
[8] A. Rimmel et al. "Current Frontiers in Computer Go," in *IEEE Transactions on Computational Intelligence and Artificial Intelligence in Games*, vol. 2, no. 4, pp. 229-238, 2010.
[9] D. Rumelhart, J. McClelland. *Parallel Distributed Processing*. Cambridge, MA: The MIT Press, 1988.
[10] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Saddle River, NJ: Pearson, 2003.
[11] I. Sutskever and V. Nair. "Mimicking Go Experts with Convolutional Neural Networks," in *Artificial Neural Networks - ICANN 2008*, 1st ed., vol. 5164. V. Kurková, R. Neruda, J. Koutník., Ed. Berlin: Springer, 2008, pp.101-110.

COAST: An architecture based on negotiation among competitive agents for automated asset management

Paulo André Lima de Castro[1], Jaime Simão Sichman[2]
(1) Technological Institute of Aeronautics - ITA
São José dos Campos , SP, Brazil
pauloac@ita.br
(2) Intelligent Techniques Laboratory, University of São Paulo
São Paulo, SP, Brazil
jaime.sichman@poli.usp.br

Abstract

*In order to manage their portfolios in stock markets, often human traders use a set of algorithms and/or indicators, which are based on stock prices series. These algorithms are usually referred to as technical analysis. However, traders prefer to use a combination of various algorithms, rather than choosing a single one: the several signals provided by these algorithms and their own knowledge are combined to determine the orders to buy or sell some stocks. Inspired by the human traders´ decision processes, our architectural approach composes heterogeneous autonomous trader agents in a competitive multiagent system. This architecture allows the use of various algorithms, based on different technical analysis indexes to manage portfolios. This architecture is named COAST (**CO**mpetitive **A**gent **S**ocie**T**y) and it is composed by two kinds of agents:* advisors *that analyze the market situation autonomously and competing with each other for resources; and* coaches *that are able to coordinate several advisors and negotiate with each other to define the best money allocation within the society. This negotiation is performed using a negotiation protocol proposed in this work. We have implemented a system using COAST architecture, using a financial market simulator called AgEx. This system was tested using real data from the Brazilian stock exchange. The test results have shown a good performance when compared to an adjusted market index. The negotiation protocol used by coach agents provided a mechanism to easily integrate new trading algorithms, without the notion of a central agent or a centralized decision mechanism, which is a highly desirable feature in scalable multiagent systems.*

1 Introduction

Multiagent systems have been used in many real problems, such as business management workflow, information management, electronic commerce, air traffic control and social simulation among others (Wooldridge, 2002). Multiagent approaches are especially interesting in problems that are naturally distributed, complex and dynamic where autonomous entities (agents) can handle some aspect of the problem and they act in a cooperative way to achieve common goals.

In this paper, we intend to use a multiagent approach in automated asset management, which is a particular case of these distributed, complex and dynamic environments. We propose here a multiagent architecture, COAST (**CO**mpetitive **A**gent **S**ocie**T**y), which is based on *competitive agents* that act autonomously on behalf of an investor in financial asset management. There are two classes of agents in the architecture. The first one composed of *competitive* agents and the second one composed of *partially cooperative* agents. Each one of the agents of the first class, called *advisors*, is able to manage one single asset according to one particular trading strategy defined a priori. The goal of advisor is to keep the highest possible evaluation from the point of view of his coach. Despite the fact that agents make their own decision independently of others, the control that one keeps over society's resources is limited and shared. Agents that belong to the second class, called *coaches*, are responsible to manage one single asset and hence are in charge of evaluating their advisors performance. The goal of each coach is to maximize the return of the investor's portfolio. Such evaluation is used in a resource distribution process, which is negotiated among the coaches. COAST architecture tries to allocate more resources to the agents with better

performance.

The rest of the paper is organized as follows. In section 2, the main concepts related to automated asset management are presented, as well as previous multiagent approaches in this domain. The main contributions of the work, the COAST architecture is described in section 3. The experiments that we have performed with the architecture are described and analyzed in section 4. Finally, we present in section 5 our conclusions and further work.

2 Automated Asset Management

The ultimate goal of an asset manager, automated or not, is to find out and adopt the most desirable set of assets for an investor, according to his preferences. The manager may adopt one set of assets through the *submission of buy and sell orders* to the stock market. The buy and sell transactions and price formation are defined through the processing of the orders of all investors in the market. In this section, we briefly present two types of analyses used in asset management: technical and fundamentalist analyses (section 2.1) and describe some related work in the automated asset management domain (section 2.2).

2.1 Technical and Fundamentalist Analyses

In the asset management domain, there are many analytic strategies based on time series analysis, which are often grouped in an approach called *technical analysis*. These strategies use some market information to identify patterns and to define orders. Some examples are *moving average*, *moving average converge-divergence*, *stochastic* and *relative strength index (RSI)*, but there are many others strategies (Castro and Sichman, 2009).

Another approach to trading strategies is called *fundamentalist analysis*. It is based on information related to economic fundamentals (including company, sector and macroeconomic fundamentals), such as net profit, market share, revenues, sector growing rates, global growing rate among others. The fundamentalist analysis approach is less used in automated asset management, despite the fact it is widely used by human asset managers. This choice is due to the greater complexity to represent in an algorithm many fundamentalist concepts, while it is quite easier to design algorithms to calculate time series used in technical analysis (Araújo and Castro, 2010).

Even within technical analysis, the identification of which information is really used and how the deliberation process occurs may change dramatically among different strategies. Furthermore, strategies may present very different performance according to market scenario (Castro and Sichman, 2007). This observation brought one first guideline to our architecture, i.e., to facilitate the composition of different strategies, like shown in section 3.

2.2 Related Work

Automated asset management, also known as automated stock trading, algorithm trading, high frequency trading and some others terms, has been a focus for many researchers (Decker et al., 1997; Yuan Luo and Davis, 2002; Kearns and Ortiz, 2003; Sherstov and Stone, 2004; Kendall and Su, 2003; Feng and Jo, 2003). It is possible to identify two different groups, according to the typical time interval between orders (or position-holding period). Strategies that have to deal with short time intervals, like weeks, few days or even fractions of second cannot be based on fundamentalist analysis, because this latter is focused on long period scenario and it is can be used only when the typical time between orders are months or years. Therefore, technical analysis is widely used for short time intervals. When the holding-position period is very short, less than one day, it is often called high frequency trading. Many researchers and practitioners have been developing algorithms to achieve better performance exploring the fact that an automated system can analyze a significant higher amount of information when compared to a human being in small time periods (Aldridge, 2009; Durbin, 2010). As far as we know, few initiatives try to explore complementarity among trading algorithms (one exception is (Castro and Sichman, 2007)), and as long we know none try to explore complementarity through a negotiated process, as it is proposed in this paper.

3 COAST Architecture

The COAST architecture is designed to facilitate the simultaneous use of many trading strategies and to explore the competition among these strategies, in order to achieve better results to the whole society. These strategies are materialized through agents called *advisors*. In COAST, strategies outputs are not interpreted as orders, but as advices about one specific asset. The other architectural guidelines are the following: **(i)** it should work with many different assets, **(ii)** it should adapt strategies' relevance for each asset and **(iii)** it should avoid central agents or a centralized decision making procedure about resource allocation through assets. In fact, there are multiple coordinator agents, called *coaches*, that negotiate among themselves and try to achieve the best possible performance to the investor. Each coach is specialized in one single specific asset. Therefore, a society with four assets and three different strategies would be composed by four coaches and twelve advisors (three advisors for each coach), like shown in figure 1. The advisors located in a same column operate with the same asset

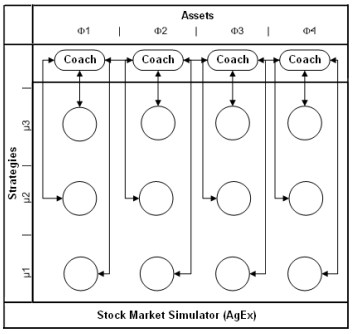


Figure 1. Example of a COmpetitive Agent SocieTy that manages four assets composing three strategies.

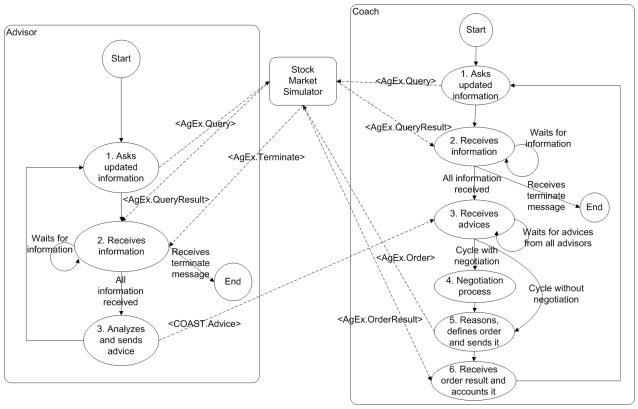


Figure 2. Advisor and coach life cycles in COAST architecture.

and the coach in top of the column evaluates and coordinates the work of the advisors in the column. According to these evaluations, one advisor with good performance has more relevance in coach decisions than the other advisors. Coaches auto-evaluate and negotiate among themselves to allocate more money to the coaches with better performance in the society, as described in section 3.3. It is important to notice that we model the architecture considering **autonomous** agents acting as experts for a specific asset, namely the coaches. Therefore, there is no central agent that controls the other agents. Indeed, coaches need to negotiate to solve conflicts and to work together.

3.1 Advisors

Advisors suggest to buy or to sell a number of shares of a specific asset following their own strategy and their goal is to give the best possible advices to improve portfolio return. Advisors can be easily created using any well known trading strategy. These advices are sent to the coach, who is the agent in charge of order definition. The advisor's life cycle is presented on the left of figure 2. In this figure, dashed lines show messages exchanged between agents and solid lines show state changes for each agent. Each state is represented by an ellipse, and has the following meaning :

1. **Asks for updated information**: The advisor, according to its strategy, asks for updated information from the *stock market simulator* (Castro and Sichman, 2009), which can be seen in the center of figure 2.

2. **Receives information**: The stock market simulator returns the information which is locally stored. This step is also used to synchronize all the agents, in simulated time.

3. **Analyses and sends advice**: According to the collected information and his strategy, the advisor defines and sends a buy/sell/hold advice to his coach.

The advisors performances are evaluated by their coach according to their advices and the market evolution. For instance, whenever an advisor suggests buying an asset whose price arises after the advice, this advisor is positively evaluated. A similar reasoning can be made regarding a selling advice.

3.2 Coaches

Coaches basically *receive* advices, *evaluate* their advisors, *negotiate* with others coaches and *define orders* that are submitted to the market. These activities are performed along all the agent lifecycle. However, the negotiation process does not happen in all cycles, only at periodic intervals which include several cycles. Negotiation in all cycles would be senseless, since the previous negotiated allocation would not have had enough time to be tested. This negotiation period is one of COAST society parameters. The coach activities are presented on the right of figure 2, and have the following meaning:

1. **Asks for updated information**: The coach asks for new information about its target asset. Even if coach orders are completely based on the messages sent by its advisors, he would need asset information to evaluate them.

2. **Receives information**: The *stock market simulator* (Castro and Sichman, 2009) returns the stored information to be used on advisors' evaluation.

3. **Receives advices**: The coach receives advices from all advisors that deal with the same particular asset that he manages. This activity is kept until he receives advice from all his advisors. During this activity, he also

R1. **If** *Advice* **is** *Buy* **and** *Evaluation* **is** *High*
Then *Expectation* **is** *Strong bullish*
R2. **If** *Advice* **is** *Sell* **and** *Evaluation* **is** *High*
Then *Expectation* **is** *Strong bearish*
R3. **If** *Advice* **is** *Buy* **and** *Evaluation* **is** *Medium*
Then *Expectation* **is** *Bullish*
R4. **If** *Advice* **is** *Sell* **and** *Evaluation* **is** *Medium*
Then *Expectation* **is** *Bearish*
R5. **If** *Advice* **is** *Manter* **Then** *Expectation* **is** *Unbiased*
R6. **If** *Evaluation* **is** *Low* **Then** *Expectation* **is** *Unbiased*

Figure 3. Coach expectation definition rules.

performs the advisors evaluation, as described in section 3.1. Whether the current cycle must include negotiation, next step is activity 4, otherwise next step is activity 5.

4. **Negotiation Process**: The coach performs negotiation with others coaches in order to define a new resource (money) allocation. This activity is much more complex than the others and therefore is described separately in more detail in section 3.3.

5. **Reasons, defines orders and sends them**: Based on advisors suggestions and following a set of fuzzy rules, the coach defines its order and sends it for execution to the *stock market simulator* (Castro and Sichman, 2009).

6. **Receives order result and accounts it**: The coach receives the order result, whose value may be total, partial or not executed at all, and accounts it in his portfolio, including the real price that was used to buy or sell the asset.

Coaches also calculate their expectation about their own performance in the near future. This performance **expectation** is modeled as a linguistic variable with five terms: *strong bearish*, *bearish*, *unbiased*, *bullish* and *strong bullish* (Pedrycz and Gomide, 1998). The coach expectation formation is based on the information (advice) that comes from his advisors and their respective evaluation. The **advice** is also a linguistic variable, with three linguistic terms: *sell*, *hold*, and *buy*. Finally, advisor **evaluation** is also a linguistic variable and have three linguistic terms: *low*, *medium* and *high*, where the universe of discourse is the success rate of the advisor [0%-100%]. Each coach receives advices from its advisors and follows the fuzzy rules presented in figure 3, in order to define its expectation.

The coach expectation is used in **order definition** and also in the negotiation mechanism when a coach must decide whether he should transfer some of his current resources to a coach with better performance. As described in section 3.3. The coach defines his order according to his expectation using a fuzzy decoding method (in our implementation, center of gravity method). For instance, a *strong bullish* market leads to a buy order with high volume, meanwhile a *strong bearish* expectation leads to a sell order with high volume and a *unbiased* expectation makes the coach to keep its current position.

3.3 Negotiation Mechanism

A negotiation mechanism is defined by a *negotiation protocol*, composed by the *communication rules* among participants, and by the players *strategies* (Rosenschein and Zlotkin, 1994). Coaches have individual and social preferences and negotiate guided by those preferences. In this section, we describe first these individual and social preferences, then the proposed negotiation protocol and finally the roles that a coach can assume in each negotiation round.

3.3.1 Individual and Social Preferences

Coaches are partially cooperative because they have a global social goal to achieve: maximize the return of society portfolio, but they try to overcome the others and get more resources to themselves, therefore they are also partially competitive agents. We believe that this competitive goal is not only acceptable, but also useful for the society, because it induces their agents to improve themselves and hence to improve the performance of the whole society. We define a utility function to represent these individual and social preferences, but some previous definitions are needed.

The return of a coach i in a time t is defined as $R(\omega_i, t) = \frac{V(\omega_i,t)-V(\omega_i,t-1)}{V(\omega_i,t-1)}$. The expression ω_i refers to resources (money and assets) allocated to agent i and function V gives its monetary value, according to the current price. The allocation ω_i may be defined as a tuple $<m_i, \omega_i^1, \omega_i^2, ..., \omega_i^n>$, where $m_i \in \Re$ defines the amount of money allocated to agent i and expression ω_i^j represents the integer number of shares of asset j held by agent i. The monetary value of whole society $V(\omega, t)$ is given by the sum of $V(\omega_i, t)$ of all the coaches $i \in C$ in that society.

Each coach has an expectation about its performance in the near future and this expectation is restricted to the interval $[O_b, O_a]$. We normalize this expectation value to [-1,1], i.e. $-1 \leq Exp_i \leq 1$. The normalized expectation (Exp_i) is used to calculate the expected monetary value for each coach. This expected value is very important to the utility function, because if one agent believes that it will have bad performance it will accept more easily to transfer his resources to other coaches.

We define the expected monetary value $V_e(\omega_i, t)$ for an agent i as $V_e(\omega_i, t) = (1 + Exp_i) * V(\omega_i, t)$. The expected

monetary value of the whole society $V_e(\omega, t)$ is given by the sum of $V_e(\omega_i, t)$ of all the coaches $i \in C$ in that society.

As coaches have individual and social preferences, they need to compose both portions to form their utility functions. The relative weight between these portions is modeled as parameter α, where $\alpha \in (0, 1)$ (zero and one are excluded), called *individuality factor*. The value (1-α) is called *social factor*. Whenever $\alpha = 1$, the agent cares only about its own goals, and would be completely individualist. Moreover, the bigger α the less concerned about social preferences would be the agent. In COAST, all coaches are concerned with both criteria, therefore $0 < \alpha < 1$.

We define the utility function $Util_i(\dot{\omega}, t)$ of a coach i as a sum of individual and social preferences weighted by its individual factor. As the negotiation process deals with resource allocation among coaches, utility functions have as parameters the proposed allocation ($\dot{\omega}$), the current allocation (implicit parameter) and a defined instant of time t, as $Util_i(\dot{\omega}, t) = \alpha * UI(\dot{\omega_i}, t) + (1 - \alpha) * US(\dot{\omega}, t)$.

The term $UI(\dot{\omega_i}, t)$ refers to the individual portion of coach preferences. It may be defined as the difference about the expected value of the new allocation and the value of the current allocation, $UI(\dot{\omega_i}, t) = V_e(\dot{\omega_i}, t) - V(\omega_i, t)$.

On the other hand, the social goal is to maximize the return of the whole society portfolio. Therefore, we define one different function to represent the social portion of coach preferences. As the negotiation deals with resource allocation among coaches, the preference function informs if a new allocation $\dot{\omega}$ is preferable over the current allocation ω, according to the social goal. This function is defined as $US(\dot{\omega}, t) = V_e(\dot{\omega}, t) - V(\omega, t)$. Hence, it gives higher numbers for allocations that contribute more to the social goal in a defined instant of time t.

By design, whenever a coach i receives a negotiation proposal for a new allocation, it will accept the proposal if his expected utility is greater or equal to zero, i.e., if $Util_i(\dot{\omega}, t) \geq 0$. The negotiation protocol is described next.

3.3.2 Negotiation Protocol

The negotiation process presented in the coach life cycle (activity 4 on the right of figure 2) is composed by seven sub-activities, which are shown in detail in figure 4. Several coaches interact with each other along the negotiation process, but there is one who is considered the coach with the best performance, who will be named *best coach*. Excluding the best coach, the *bad coaches* are all coaches with negative expectation and the *neutral coaches* are those with positive expectation.

The negotiation process sub-activities are the following :

4.1. **Sends information for others coaches**: Each coach sends to his acquaintances information about its own performance (risk, return and patrimony) and expectation about the near future.

4.2. **Receives information from other coaches**: Each coach receives information from all the others. Hence, each one may calculate the society patrimony, risk and return.

4.3. **Defines coach roles according to performance**: In this activity, each coach calculates the possible roles that each coach, including himself, can play (best, bad or neutral). The coach roles definition is performed by each coach separately, because they are completely autonomous and do not have precedence over the others. However, since we consider that coaches do not lie to each other, they all achieve the same result, since they use the same information. The best coach executes activities 4.6 and 4.7; the others execute activities 4.4 and 4.5.

4.4. ***Other coaches*** - **Receives and analyzes proposals**. This analysis is performed according to individual and social preferences and the current observed situation. Each agent decides if he should accept the proposal or not, based on his utility function as explained in section 3.3.1.

4.5. ***Other coaches*** - **Sends proposals answers**: The proposal answer is sent back to the proponent. If the answer is affirmative, the new allocation is adopted.

4.6. ***Best coach*** - **Prepares and sends proposals**: The best coach prepares a proposal that asks for all available money from the bad coaches.

4.7. ***Best coach*** - **Process proposals answers**: The best coach waits for all the proposal answers. Each affirmative answer creates a deal and the transfer is performed immediately. In case of a negative answer, nothing is changed and the proponent agent does not receive any money from the agent that refused the proposal.

4 Experiments and Results

We have implemented a version of COAST architecture that uses four advisors strategies based on technical analysis. The technical indexes used are the following: moving average, moving average converge-divergence, relative strength index and price oscillator, mentioned in section 2. We have selected 15 assets, which are part of the main index in Brazilian stock market, IBOVESPA, and presented a big number of trading in the last five years (from January, 2006 until December, 2010). Simulation experiments have been performed using a market simulator called AgEx (Castro

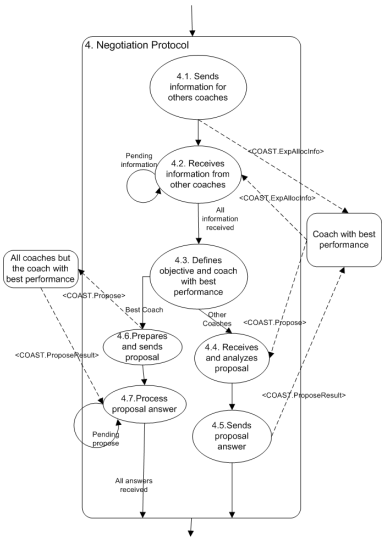


Figure 4. Negotiation process among coaches in COAST architecture.

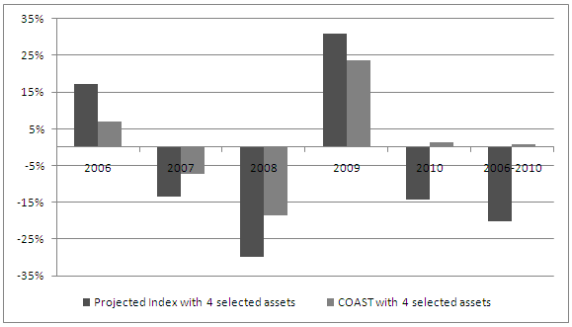


Figure 5. Return achieved by COAST and *Projected Index* using four selected assets

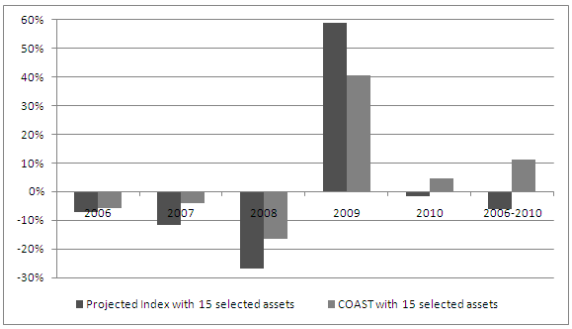


Figure 6. Return achieved by COAST and *Projected Index* using fifteen selected assets

and Sichman, 2009). We tested COAST societies trading in exchange using daily quotes, where each coach could give one order a day. Additionally, we despised the effect of the orders given by the coaches in the market price, because the agents deal with a very small amount of money when compared to the traded volume for each asset. Despite the fact that the market simulator allows the use of transaction fees, for simplicity we set these fees to zero. In fact, transaction fees have small influence on performance, since there is no big difference in the number of orders given by the analyzed societies (Castro and Sichman, 2009).

In order to analyze COAST performance, we have executed simulation experiments using two different COAST societies, the first with four assets and the second one with all fifteen selected assets. Our first idea of direct comparison was to use the IBOVESPA index. However, a comparison among COAST performance and IBOVESPA would be biased because they do not deal with the same assets. In fact, IBOVESPA index composition changes in time and many assets have been included or excluded along the five years of the evaluation period, i.e. from 2006 to 2010. Due to these facts, we have created a theoretical portfolio called *Projected Index*, which is composed by the fifteen assets used by COAST societies, according to the relative weight of each asset (p_i). Moreover, we normalized these weights to use only the chosen assets using $p_i = \frac{w_i}{\sum_{j \in PI} w_j} * 100\%$, where p_i is the asset weight in Projected Index and w_i is the original weight in IBOVESPA. We have used these weights to define an AgEx trader agent (Castro and Sichman, 2009), which buys and holds a set of shares according to the specified weights. This agent, called also *Projected Index*, acted in the same simulated evaluation period of five years. The figure 5 shows the performance of a COAST society managing four assets against Projected Index with the same four assets. In figure 6, we present the performance of COAST and Projected Index when managing all fifteen assets and finally the performance comparison among the two COAST societies are presented in figure 7.

The comparison among the COAST societies and the *Projected Index* agents in return (figure 5 and 6) shows a better performance of COAST in most of the years and in the whole period (2006-2010). In figure 7, we compare the performance of the COAST societies, the first dealing with four assets and the second with fifteen assets. It is easy to notice that the society with bigger number of assets presented a better performance in all years, except for 2006, and the best performance in the whole period. These facts make us believe that it may be possible to pursuit better results with more assets and that is possible to achieve good performance in using a competitive agent approach.

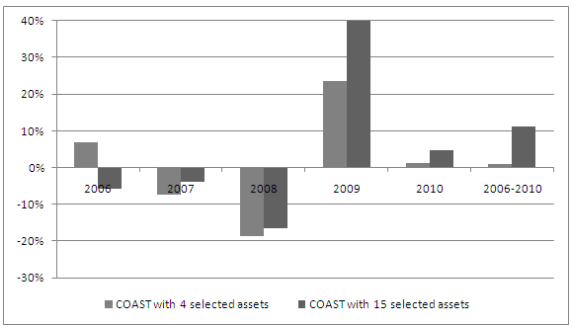


Figure 7. Return achieved by two COAST societies with four and fifteen assets

5 Conclusions and Further Work

In this paper, we presented the COAST multiagent architecture, their agents and a new negotiation mechanism specially designed for this architecture. COAST architecture was implemented and tested in several simulation experiments, which results were presented and analyzed. In the simulated experiments, COAST architecture showed good results and overcome in some scenarios the chosen benchmark (*Projected Index*). It was also possible to realize that is possible to achieve better results using more assets in society.

The main contributions of this work are the *exploitation of competitive strategies* within the COAST architecture (section 3), the fusion of competitive strategies through fuzzy logic and the proposed negotiation mechanism (section 3.3). It also facilitates the use of well known trading strategies as advisors agent's strategies.

In future work, we intend to test COAST architecture with more trading strategies and using a wider evaluation period and number of assets. We believe that a significantly evolution would be a formal modeling of expectations, which are very important in economic reasoning.

Acknowledgments

Jaime Sichman is partially funded by CNPq and FAPESP, Brazil. Paulo Castro thanks Prof. Alexander Brodsky from George Mason University for the basic idea of the *Projected Index* agent.

References

Aldridge, I. (2009). *High-Frequency Trading: A Practical Guide to Algorithmic Strategies and Trading Systems*. New York: Wiley.

Araújo, C. H. D. and P. A. Castro (2010). Towards automated trading based on fundamentalist and technical data. In *Proceedings of 20 th SBIA. LNAI*, São Bernado do Campo, Brazil, pp. 704–715. Springer-Verlag.

Castro, P. A. and J. S. Sichman (2007). Towards cooperation among competitive trader agents. In *Proceedings of 9th ICEIS.*, Funchal, Portugal, pp. 138–143.

Castro, P. A. and J. S. Sichman (2009). Agex: A financial market simulation tool for software agents. In W. Aalst, J. Mylopoulos, N. M. Sadeh, M. J. Shaw, C. Szyperski, J. Filipe, and J. Cordeiro (Eds.), *LNBIP*, Volume 24, pp. 704–715. Berlin: Springer.

Decker, K., A. Pannu, K. Sycara, and M. Williamson (1997, 5–8,). Designing behaviors for information agents. In W. L. Johnson and B. Hayes-Roth (Eds.), *Proceedings of the First International Conference on Autonomous Agents (Agents'97)*, New York, pp. 404–412. ACM Press.

Durbin, M. (2010). *All About High-Frequency Trading*. New York: McGraw-Hill.

Feng, X. and C.-H. Jo (2003, 3). Agent-based stock trading. In *proceedings of the ISCA CATA-2003*, Honolulu, USA.

Kearns, M. and L. Ortiz (2003, 11-12). The penn-lehman automated trading project. *IEEE Intelligent System 18*(6), 22–31.

Kendall, G. and Y. Su (2003, 6). Co-evolution of successful trading strategies in a simulated stock market. In *proceedings of The 2003 International Conference on Machine Learning and Applications (ICMLA'03)*, Los Angeles, pp. 200–206.

Pedrycz, W. and F. Gomide (1998). *An introduction to fuzzy sets: analysis and design*. Cambridge, Massachusetts: The MIT Press.

Rosenschein, J. S. and G. Zlotkin (1994). *Rules of encounter:designing conventions for automated negotiation among computers*. Boston: MIT Press.

Sherstov, A. and P. Stone (2004). Three automated stock-trading agents: A comparative study. In *Proceedings of the Agent Mediated Electronic Commerce (AMEC) Workshop - AAMAS 2004*, New York.

Wooldridge, M. (2002). *An Introduction to MultiAgent Systems*. London, England: Wiley.

Yuan Luo, K. L. and D. N. Davis (2002). A multi-agent decision support system for stock trading. *IEEE Network 16*(1), 20–27.

USING MULTIAGENT PLANNING IN THE HOSHIMI PROJECT

B. Nepomuceno[1], **Carlos R. Lopes**[2]
[1]University Center of Patos de Minas, Patos de Minas, Minas Gerais, Brazil
[2]Faculty of Computing, Federal University of Uberlândia, Uberlândia, Minas Gerais, Brazil

Abstract— *With a digital game industry in constant ascendant and a more demanding consumer, designers and programmers have faced even bigger challenges to create realistic games. This paper presents an analysis of use of a planning algorithm based on A* and STRIPS in an environment called Hoshimi Project. It also presents two simple and efficient multiagent planners.*

Keywords: Multiagents, Planning, Artificial Intelligence, Hoshimi Project

1. Introduction

Trying to create more realistic games, developers frequently find challenges that can be overcome by academic theories [1]. The computer games domain, characterized by complexity and dynamism, is an excellent lab to new researches, mainly those involved with Artificial Intelligence [1]. For example, when an NPC (Non-Player Character) needs to explore a new world, it uses, frequently, an A* algorithm or one of its variations. The game industry grows considerably every year. There are more games in the market, but the main characteristic is that they always are more realistic and the player is always more demanding. Thus, old techniques like FSM (Finite State Machines) and RBS (Rule Based Systems) used to define the behavior of an NPC are being replaced. Instead of traversing a predefined graph of state transitions, an NPC controlled by a planning algorithm searches for a sequence of actions to satisfy some goal [2]. This is the idea behind the awarded game F.E.A.R by Monolith Productions [3].

By planning techniques it is possible to create an NPC more flexible and capable of manage unexpected situations. Besides, using this system of goals and actions helps to make blocks of code easier to maintain and re-use [4]. Because of this, FSM are being replaced by a GOAP (Goal-Orient Action Planning) architecture. Planning techniques are implicitly related to resource management. In an RTS (*Real Time Strategy*) game, players must manage resources like gold, wood, food and use it to improve their army or characters skills [5].

However, typically, it is hard to get a good result in terms of performance when working with planning algorithms. This situation is intensified when the environment is complex and dynamic like computer game domain. Therefore, it is necessary to use an efficient knowledge system and an algorithm that can take advantage of these generating plans within an acceptable time. This paper proposes to verify the viability of using F.E.A.R planning algorithm in another kind of environment rather than FPS (*First Person Shooter*), an RTS game like called Hoshimi Project. Moreover, we present two multiagent planning algorithms, which are efficient in the computer game domain.

2. The Hoshimi Project

The Hoshimi Project was the environment chosen to test agents architecture and planning algorithms [6]. It was used in a Programming Battle category of Imagine Cup, a worldwide contest created by Microsoft Corporation. This environment was used in 2005, 2006 and 2007 edition of Imagine Cup.

In the Hoshimi Project, the programmer may use a big variety of classes to control a set of nanobots. These nanobots are virtually injected inside a human body to cure diseases. There are lots of things to be done in Project Hoshimi. First, the programmer must create an AI to control this set of nanobots and make them achieve some objectives in different parts of a human body. Each part is a map and has its own objectives. There are many kinds of objectives like: exploration, survival and collect and transfer of enzymes. The maps have points with special functions. The most important are AZN and HP points. An AZN point is where a nanobot (nanocollector or nanocontainer) collects enzymes and a HP point is where a nanobot deposit enzymes to cure the body and accomplish objectives. When enzymes are deposited in a HP point, the team of bots makes points. Thus, no matter the map, collect and deposit enzymes is always something good to do in Hoshimi Project environment. There are a lot of details that are not relevant for the understanding of this paper like density of cells and blood streams that can be found on the official site of the contest [6].

3. Architecture

The architecture of the agents used in this work was based on MIT Media Lab C4 [7]. In this architecture an agent is composed by a memory unit, perception, planning and executing actions components, and a system based on a blackboard is used to exchange information between the architecture components. When the sub-system responsible for defining the next action to be executed (Planning sub-system) wants to communicate with the motor system to tell the position the agent must go, it uses the blackboard. Its a

simple idea: the blackboard has many attributes that can hold values necessary for sub-systems. Hence, the only thing the planner sub-system must do is to write the position value in the proper attribute. All goals are placed there, too. In the same way, when the sub-system responsible for perceiving the environment wants to generate another goal, it uses the blackboard to tell it to planner sub-system.

4. The Planner

A planning problem requires three inputs: the initial state of the world, a goal and a set of actions that can be used to achieve goals [8]. The result of a planning process is a sequence of actions that when executed over the initial state, achieve a goal [9]. The sub-system created in this work to solve planning problems was called *AStarPlanner* and was based on Jeff Orkins work [2], [4] where he uses an A* algorithm [10] to generate plans. Generally, this algorithm is used to solve path finding problems, but in fact, the process is very similar. Both start in an initial state and through the execution of a set of valid operations, produce a desired state. One of the main objectives of this work was to verify the viability of using Orkins algorithm in Project Hoshimi context. In a path finding problem, a graph node generated by A* maps a 2D or 3D coordinate. What links a node to another is a movement operation. Talking about plans, the generated node corresponds to a world state and what links a node to another is an action (ex.: move to some location, collect some enzymes, shoot).

Its obvious that its not right to represent every piece of information about a world state in a node used by A*. This would generate algorithms with low performance. On the other hand, each node must have enough information to generate good plans. The solution is to create a knowledge system based on symbolic representation centered in the agent. In consequence, the whole knowledge that the agent knows is about itself, there is no need to discover about what object information is about. Every piece of information would be stored in a *WorldStateProperty* object that has: a symbol, a data type and a value. The symbol represents the kind of knowledge stored. For example: location, location type (perhaps AZN or HP points), indicative that a nanoneedle is full or not. The action representation system is very similar to action representation in STRIPS [11]. But, instead of using a list of facts that will be added or removed from the world, it is used a list of *WorldStateProperty* objects. So, the system has two lists of *WorldStateProperty* objects: one to hold the preconditions of an action and another to hold its effects. An action also has some methods: one to calculate its cost of execution, one to perform dynamic calculations and another to specify the real behavior of the agent (the one which will occur when the agent executes the action). The method for performing dynamic calculations is not present in Orkins implementation. As mentioned before, the list of effects is stored using *WorldStateProperty* objects, and for each property there is a value. Some actions have effects that have dynamic values. For example, consider a *GotoType* action, an action that sends one nanobot to one specific type of location. Its primarily goal is to change the type of the location of the bot but, consequently the location of the bot will change too. The value of this new location is calculated in the dynamic calculations method. To work with A*, its necessary to define a cost and a heuristic. The cost of each action was defined as 1. However, if it were interesting to specify kind of a priority system of execution, it could be done using different values for actions costs. For instance, in a FPS game, it is more interesting to shoot other characters from places that can provide some kind of protection. In the beginning, when was important to discover if a planner based on Orkins ideas would be useful in Hoshimi Project environment, the number of unsatisfied symbols was used as the heuristic. But, when the multiagent planners were implemented, the amount of time to execute one action was used instead.

The algorithm was tested for all kinds of Hoshimi Project objectives and was succeeded in every one. It always generated plans to achieve the existing goals. One important fact is that all plans generated have low cost of execution, once the algorithm is based on A* that finds the best path between two points, in this case, between two world states.

5. Multiagent Planning

After discovering the viability of an algorithm based on A* and in the symbolic knowledge system, the next step was create algorithms to solve planning problems using many agents and based on *AStarPlanner*. Two kind of algorithms were proposed and tested.

The first algorithm used a notion of an auction similar to one described by [12]. This version was called *AuctionAStarPlanner*. Whenever a goal was generated, an auction was presented to nanobots. To reduce the amount of nanobots that could take part of the auction, was made a selection based on the effect list of actions of each nanobot. Those who had actions with list of effects with symbols that appear on the goal could participate of the auction; the other were dismissed. The agent that presents the plan with the lower time of execution wins the auction. Its important to say that one nanobot can take part of many auctions at the same time. Thus, if a nanobot take part of two or three auctions, the time necessary to execute the plans that he already won would be considered as the state of the world generated after the execution of prior plans.

The second algorithm is very simple, but in Hoshimi Project context, showed a good performance. Whenever a goal is generated, one nanobot that is idle in that moment will be responsible to accomplish that task. If no one is, when the first one finishes its job, it will assume that goal. In the case of two or three idle nanobots, the first one to communicate its idleness gets the goal. There is a problem:

as some plans are relatively big for this kind of domain (FPS or RTS games), one goal could be break in two or more sub-goals. Hence, its possible to have, in this version, two or more nanobots working in the same goal. This version was called *SimpleMultiAgentPlanner*.

6. Experiments

Once we have tested and verified the effectiveness of the algorithm based on A*, the tests with the multiagent planners were performed only with one kind of objective: collect and transfer AZN. This decision was taken to make comparison between algorithms easier.

Six maps were chosen to be used in the tests in order to have a case test with maps presenting different characteristics. Four maps were used in the international final of Imagine Cup of 2007 and the other two were chosen because they had particular and extreme scenarios. The SC1 map was chosen because it has few obstacles and makes movement actions easy tasks and contributes to a good performance on algorithms that estimate the cost of moving from one point to another using Manhattan distance. This can be seen in Figure 1.

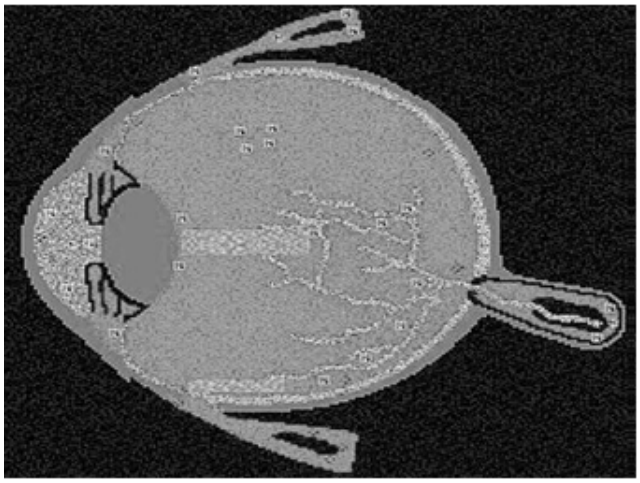


Fig. 1: SC1 Map.

In Figure 1, there is a quite opposite situation; there are many obstacles and an absence of a contiguous terrain

To transfer enzymes to a HP point is necessary that a special nanobot called *AI* build a nanobot called *nanoneedle* in the HP location. Thus, it is necessary that this AI bot traverse a sequence of HP points. In the tests, the routes used were fixed; every time the program runs, the sequence of points to be traversed is the same. To create a more interesting test, we created two kinds of routes: one totally random and another optimized. In the beginning of the game, some goals are generated and after some period of time, other ones are generated. We tested systems with sets of 3 and 6 nanobots. During the game, it is not very common to use more than 6 nanobots to act as collectors because there are other goals that require other kinds of bots and there is a total number of bots that can be used during the game.

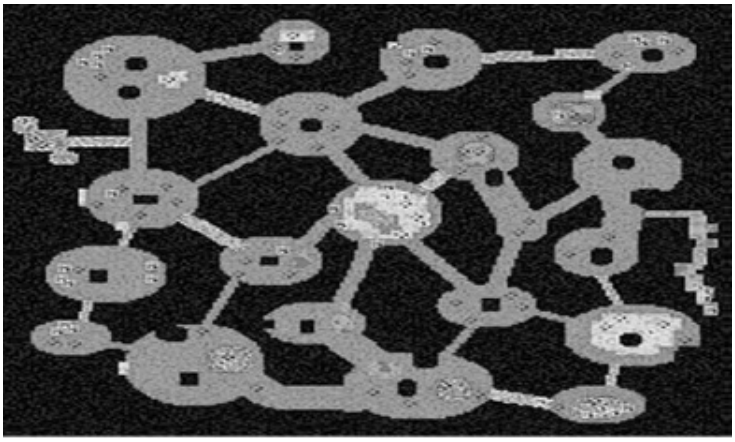


Fig. 2: SC3 Map.

In a general way, both algorithms presented practically the same results. Initially, the authors expected that the one based on the auction system would have a better performance. But, that feature of breaking goals in sub-goals helped a lot the *SimpleMultiPlanner* algorithm.

Figure 3 presents a table of the points obtained by each planner in the six maps (whenever a bot achieve an objective it gains points) Multi represents the *AuctionAStarPlanner* and Single represents *SingleMultiAgentPlanner*. OR is used to represent tests that used an optimized route and WOR those ones that didn't use optimized routes (without optimized routes). The numbers after OR and WOR indicate the number of bots used in the test (3 or 6).

Map	Single		Multi		Single		Multi	
	OR3	WOR3	OR3	WOR3	OR6	WOR6	OR6	WOR6
SC3	2850	2540	2850	2540	3845	3170	3805	2795
SC1	3235	2430	3045	2780	4415	3910	5060	4205
A1	1960	1650	2055	1180	3290	1880	3430	1180
A2	4010	3210	4115	3225	5690	3910	5880	4105
A3	1525	2275	1525	2330	3460	2450	3555	2585
A4	2215	2420	2255	2405	4295	3590	4120	3550
Sum	*15795*	*14525*	*15845*	*14460*	*24995*	*18910*	*25850*	*18420*

Fig. 3: Table of results

Figure 4 shows a comparative graphic for the first three maps.

It is important to note that, as in an auction system wins that one that generates a plan with lower time of execution, the cost of execution must be observed. To calculate the cost of actions that involve movement, Manhattan distance was

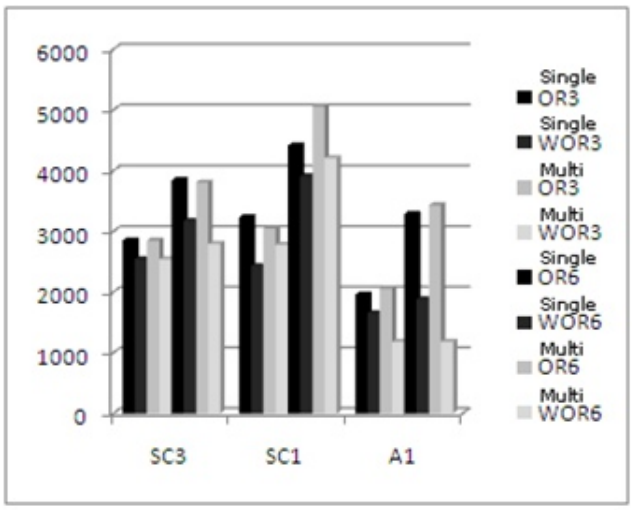


Fig. 4: Comparison of SC1, SC3 e A1 maps

used. This explains the good performance of *AuctionStarPlanner* on SC1 (Figure 1) map and not so good at SC3 map (Figure 2). Figure 5 shows a comparative graphic for the last three maps.

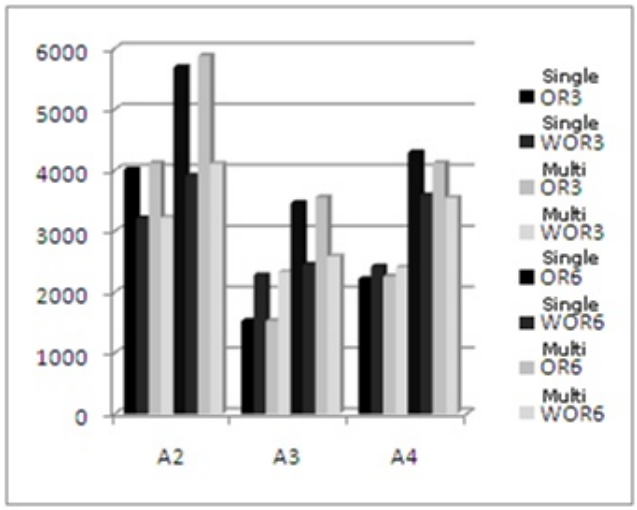


Fig. 5: Comparison of A2, A3 e A4 maps

In these three maps, *SimpleMultiAgentPlanner* had a little inferior performance, but nothing really significant. An interesting fact is that the route did not affect the results. It was expected that *AuctionStarPlanner* would have a better performance in tests where non optimized routes were used, but it did not happen.

During the tests, all the time the authors tried to identify the optimum result for each case and for all of them the plans generated by *AuctionStarPlanner* and *SimpleMultiAgentPlanner* obtained a performance level near the optimum.

7. Related Work

This work was based on the MIT researcher called Jeff Orkin [2], [4] where he uses an A* algorithm to generate plans and a notation similar to STRIPS to represent world states and actions. In his work, Orkin test his ideas on a FPS game and suggests that it could work on other kind of games. In this work, was verified that Orkins ideas could be used in Hoshimi Project environment and that the planner could be used as a part of multiagent planners with good performance.

8. Conclusion

As presented before, the algorithm used in the game F.E.A.R can be used in other kinds of games rather than FPS as Orkin suggested. Besides, it was discovered that simple ideas can be used to produce multiagent planners with good performance in computer games domain. In the future, the authors plan to implement a multiagent planner that involves many agents in the generation of a single plan.

Acknowledgments

We thank FAPEMIG, CAPES, and UNIPAM for providing financial support for the development of this research.

References

[1] B. J. H. J. H. S. S. K. T. R. LAIRD, J, "Intelligent agents in computer games," *In Proceedings of The Sixteenth National Conference on Artificial Intelligence*, pp. 160–173, 1999.

[2] J. ORKIN, "Symbolic representation of game world state: Toward real-time planning in games," *AAAI Workshop on Challenges in Game AI*, 2004.

[3] M. P. Inc, 2006. [Online]. Available: http://www.lith.com/home.asp

[4] J. ORKIN, "Agent architecture considerations for real-time planning in game," *AIIDE*, 2005.

[5] M. Buro and T. Furtak, "Rts games as test-bed for real-time research," *Workshop on Game AI, JCIS, 481-484*, 2003.

[6] HOSHIMI, "The hoshimi project," 2007. [Online]. Available: http://www.project-hoshimi.com/

[7] I. D. D. M. I. Y. BURKE, R. and B. BLUMBERG, "Creaturesmarts: The art and architecture of a virtual brain," in *Proceedings of the Game Developers Conference, San Jose, CA,*, 2001.

[8] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 2009.

[9] D. S. WELD, "Recent advances in ai planning," *AI Magazine*, vol. 20, pp. 93–123, 1999.

[10] A. J. PATEL, "Amit's thoughts on path-finding and astar," 2006. [Online]. Available: http://theory.stanford.edu/ amitp/GameProgramming/

[11] R. E. FIKES and N. J. NILSSON, "A new approach to the application of theorem proving to problem solving," *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 279–293, 1971.

[12] R. e. a. KROGT, "Multiagent planning through plan repair." *AAMAS*, 2005.

Crowds and Spontaneous Collaboration

Henry Hexmoor
Southern Illinois University
Computer Science Department
Faner 2130, Carbondale, IL, 62901, USA

Abstract - *Study of Crowd dynamics has had significant overlaps with models of biological swarms. Understanding and directing human crowds have also been of long-standing interest. In this paper, we describe a few basic cognitive processes that account for life cycle of typical human crowds. Individuals change their behavior with stimulants that attract a crowd. A crowd remains in place and behaves as a collective unit as long as stimulants persist. Once original stimulants cease or others emerge, the crowd disperses and individuals return to their individualistic behaviors. Although crowd phenomenon is commonly observed in natural settings there are no previous explanations for it. Our model stride lays the groundwork for further modeling of common human crowds.*

Keywords: Human crowd model, multiagent simulation

1 Introduction

At very high autonomy levels that are needed for machine to machine collaboration in mission critical operations of the DoD, collaboration among entities of a group such as a battalion can be seen to be in the form of shared world views. This is the perspective that entities share ordinary as well as mental states with their peers through networked medium such as a social network. Decisions and actions taken by an individual affect others. The context of a given domain affects others with varying impacts and speeds. For instance, in a typical legal setting, the time horizon of interest can be measured in days to weeks whereas in an air traffic control the time span is in seconds to a minute. In sporting teams or swarm flight formation, the time span is yet smaller. In a first responder or gunship control the time span of interest descends below milliseconds. At the latter settings, there is no time for effective human level communication or much less argumentation involving humans. For brevity, we'll dub this context Spontaneous collaboration (SC). There are natural limitations for a social network providing a shared world view for SC. Groups in a social network cannot behave as if they had a single mind that is often required for SC. Despite requiring biological conspecificity, a crowd cannot be properly modeled as a social network since individuals still operate with independent minds. Cognitively, Marvin Minsky likened the human mind to a society of agents [12]. However, the reciprocal of likening a group/society cognitive unit to a single mind is not feasible. Life cycle of a group mind is fascinating and discussed in this paper. At some point individuals yield their own reasoning to the group for a marginal gain from the collective advantage as in sheep behavior [5]. The state is maintained as long as individuals draw benefit from it. An individual departs from the pack when it senses a loss of original advantage. Ed Hutchins of UCSD has studied the psychology of group cognitive units in primates as well as among the Naval aircraft crew [7]. He initiated the subfield of distributed cognition in psychology with implications for social cognition. Drawing inspiration from social cognition, we believe that the cognitive processes responsible for initiating cognitive cohesion among individuals in a close knit group to be the foundations of crowd cognition. Homophily is an example of psychological attraction to be with others [8].

Numerous studies have been made trying to understand the collective behavior and the crowd formation of animals. Animal crowds have different names like swarms, flocks, schools, colonies and etc, depending on the specie. Researches try to understand why animal crowd and also how the animals act collectively. Numerous models have been designed to represent these crowds, models that are based on the size and the density of the crowd, which are interrelated according to Niwa's model, that the mean group size will strictly increase with population density [19].

Homo-sapiens behave collectively as well for all the same reasons as the animals do, but also for other reasons; due to our social intelligence and our ability to create crowds and groups, not only because of our natural instincts, but also willingly and purposively. This volitional behavior is creating

masses of people that have similar behavior and collective intelligence-- these masses are called crowds of people, or mass, or mob, etc, depending on the meaning and the purpose of the gathering.

The impetus that propelled ancient Greeks, as philosophers, to explore crowds and mobs were to understand and harness the power that crowds possess. They distinguished mobs, which were called ochlos (Greek: οχλος), from crowds due to the reason that mobs were easy to manipulate, easy to sway and persuade. They created the term ochlocracy that means government by a mass of people, and they discriminate ochlocracy as a bad form of government. They also created democracy, a government type that is still popular today. Ancient Greeks understood that by studying crowd behavior and crowd psychology they could find ways to manipulate them and lead them to a desired conclusion.

One of the original founders of crowd research is Gustave Le Bon, who defined crowd in his book [10]. Le Bon's crowd is any gathering of people of whatever nationality, profession, or sex, and whatever be the chances that have brought them together [10]. He defined crowds as a gathering of anonymous people that tune to the lowest level of intelligence, present in the crowd, to achieve a certain goal. According to Reicher [15], to define a crowd is more complicated and more difficult that it seems. Since events in a crowd cannot be reduced to a generic set of behaviors, Reicher argues that all the classic accounts of crowds fail to give an accurate definition [15].

We define crowds as groups of people that are identified by their shared ideas, principles, emotional experiences, behavior, and goals in a share physical and social space. Therefore, crowds possess heterogeneous set of individuals and based on their shared ideas, principles, emotional experiences, behavior, and goals, form a cohesive units with homogeneous individuals.

There are models that have been created to represent a crowd of animals that are genetically identical and the crowd formation helps them to forage for food as in ants, to save energy as in ducks, to be secure as in fishes, and also to facilitate procreation. Other models consist of group of animals that are genetically unrelated and have similarly shaped group size and live in relatively homogeneous environments, like zebras and buffalos that have the same eating habits and can move at the same speed [19].

Many animals crowd using environmental homogeneity. Diverse physical features of the physical world could simply be attracting forces, rather than aggregation in response to other animals, for example water can attract heterogeneous animal species [19]. Another possibility is that the distribution of a predator species that crowd together is simply the existence of a potential prey [19].

A model can be designed detailed enough to represent a life cycle of crowd, from formation process to dispersion. In this paper we will outline salient factors that drive the behavior of individuals to the formation of a crowd and to collective intelligence. We will model a general sense of human and crowd behavior based on psychological and engineering principles, driven by external influences, environmental constraints, time, common and individual goals, and affects. Next we describe our generic model of human crowds.

2 Sketch of a Model

If we accept the research of Allport, then individuality is more prominent than crowd behavior: There is no psychology of groups separate from psychology of individuals. The individual in the crowd behaves just as he would behave alone [1]. It would be highly unlikely that any generic model could summarize every kind of crowd behavior or even a single one. The reason is that with any added individual to the crowd, the cohesive individuality in that crowd increases in an incalculable manner. However, based on our model, we can make the assumption that the growth of the complexity of a given model is increasing up to certain point. Next, we describe the crowd life cycle. We start with individuals and we track the process that turns them into a crowd. Also, we track the crowd members as they return to their state of individuality. Each stage is described by attributes that we will outline. Each stage can be further analyzed and then can be combined with the rest. Each stage, however, has its own steps therefore can be viewed as a sub-model on its own.

According to economists a rational man is an economic man and he is assumed to have knowledge on the relevant aspects of his environment, which if not absolutely complete, is at least impressively clear and voluminous. He is assumed also to have a well-organized and stable system of preference and a skill in computation that enables him to calculate, for the alternative courses of

action that are available, which of this will permit him to reach the highest attainable point on his preference scale [17].

According to Simon, we do not try to optimize our goal but satisfy our goal. Simon said that a rational individual has to be bounded by rationality that is unique human form in which a person arrives at a solution/goal that will satisfy his or her own needs even though it may not be computably perfect [17]. Simon's view is considered to be the correct one because in the real world the person does try to achieve the optimized strategy, but the person can be satisfied with close alternatives.

Crowd Formation is the most important part of the model for the reason that understanding the Crowd Formation leads us to the understanding of Crowd and Crowd Dispersion too. In this thesis, we consider Crowd Formation and Crowd Dispersion as an opposite process. By comprehending what motivates individuals to form a Crowd, we can find a way to satisfy their goal, which will lead to the dispersion of the Crowd.

Crowds are defined as groups of people that are distinct by their shared ideas, behavior and goals. These common attributes are what generate the formation of a Crowd in the first place. We suggest that each individual communicates with one's surrounding individuals (the ones in immediate physical proximity of himself). As we mentioned before, an individual is a rational entity and that results in a centralized conception of the self. So, we can apply the idea of this centrality of the individual in a social network by representing a single individual as a central vertex, in a social space [16]. According to Forsyth the core of a crowd is the groupings. The formation of the Crowd begins from certain stimuli that lead the individuals to recognize their common goal, behavior, feeling or affect as another person [2]. Therefore, the union of the common attributes of the different individuals can create an edge between two vertices on the graph that represents the relationship. (see Figure 1).

The more common attributes that two individuals share, the stronger relationship they possess, so in the graph the line that connect the two vertices will be thicker. Human beings have the tendency to reduce their physical and social need space and form relationships with people that are very similar to them, that is why, we can distinguish stronger relationships with vertices that are very close [8].

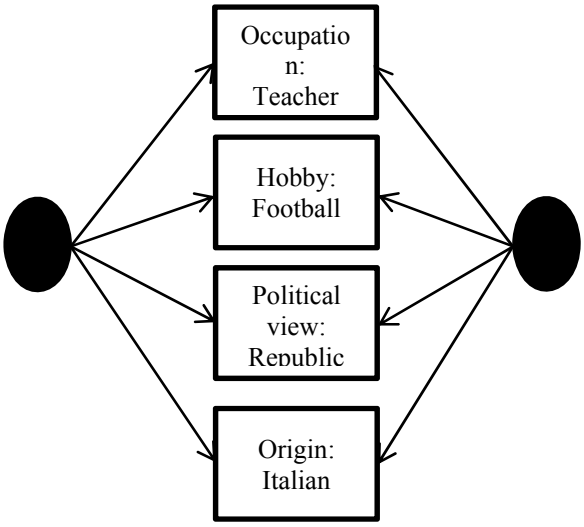


Figure 1. Possible Common attributes between individuals

When a group already exists and because of this existence, the behavior of the members of the group and the surrounding individuals is influenced. The self-categorization theory says that when people are members of a certain group, their behavior automatically changes to meet the group mentality of that group [20]. Moreover, according to this theory, for a particular member to categorize himself in a crowd, is driven by the social identity theory that involves three psychological processes.

1. Social Categorization: a way of thinking about the self and others that emphasizes membership in a group.
2. Social Identity: a portion of the self-concept that reflects the groups to which someone belongs.
3. Social Comparison: people compare their own group to any other group that seems relevant [14].

Crowd formation is simply the effort of an individual to categorize himself/herself in a group/different group, based on common attributes with the rest of the group members. The best categorization an individual can chose is the one that maximizes the similarities and minimizes the differences amongst the people in the same group/groups and also minimizes the similarities and maximizes the differences amongst the people in the out-group. This process is called meta-contrast ratio [14].

A Crowd Behavior is carefully regulated by already existing social Norms. Those Norms are providing limits for the behavior but also members

have to behave almost the same as their surrounding members in that Crowd [6]. The group behavior will result to collective judgments and collective choice. As people behave almost the same with their neighbors, they adapt even more extreme judgments than they intended [6]. Because of this behavior, most of the time, the decisions and preferences are conceptually distinct from the initial opinion of the individual [6].

Our model of the life cycle of a crowd consists of the following consecutive six stages. Stage 1 leads to 2 and on to 6 and back to 1 as such forming a complete cycle. The core of such a crowd exists between stages 2-5. This is an abstract model and describes the crowd in general. Crowds are ubiquitous and there are large class of crowd types with analyses that are outside our current scope. We envision that specific details for stages 2 and 5 that lead to formation and dispersion of a crowd can be divided into subcategories.

1. **Individual stage**: People reason autonomously and independent of others.
2. **Stimulation stage**: A group of people simulatenously experience commonality of feelings (e.g., fear) and reactions to external stimuli (e.g., flight to safety). Commonalities might include goals and behaviors (e.g., safety).
3. **Crowd Formation Stage**: The group of stimulated individuals behaves similarly as a collective unit.
4. **Crowd stage**: Once a crowd is formed in the previous stage, they remain a crowd and can be seen as a collective unit.
5. **Stimulation stage**: This stage is the opposite of stage 2 where the group experience lack of commonality of feelings and reactions to external stimuli. This divergence Commonalities might include differences in goals and behaviors.
6. **Dispersion stage**: At this stage the crowd begins to break apart its cohesiveness. This stage leads back to the individual stage.

Our model starts with individuals and we track the process that turns them into a crowd. Also, we track the crowd members as they return to their state of individuality. Each stage can be further analyzed and can be combined with the rest. The result is this model.

We have to understand the mind of a single individual first. The English dictionary presents a person as an individual human being with reference to his/her social relationship and behavioral patterns as conditioned by the culture (according to Dictionary.com). We will consider the individual as a rational entity. According to economists a rational man is an economic man and he is assumed to have knowledge on the relevant aspects of his environment, which if not absolutely complete, is at least impressively clear and voluminous [17]. He is assumed also to have a well-organized and stable system of preference and a skill in computation that enables him to calculate, for the alternative courses of action that are available, which of this will permit him to reach the highest attainable point on his preference scale [17].

Crowd formation is the most important part of the model for the reason that understanding the crowd formation leads us to the understanding of crowd and Crowd Dispersion as well. We consider crowd formation and crowd dispersion as opposite processes. By comprehending what motivates individuals to form a crowd, we can discover a way to satisfy their goal, which will lead to the dispersion of the crowd.

As we define in the introduction, crowd is a group of people that are defined by their shared ideas, behavior, goals etc. When an individual joins a crowd, the sentiments and ideas of that individual take a direction, the same direction as the rest of the crowd [3]. If we partially accept Le Bon's theory, the conscious personality vanishes and a collective mind is formed. With this in mind, we can characterize the crowd in few ways. Canetti proposed that crowd always wants to grow. For example, if an accident occurs, we can observe that the crowd around it will be getting bigger and bigger by the minute up to a certain point, which is specified by physical space and time. Secondly, within the crowd, there is a quality and thirdly the crowd loves density. The bubble space of each individual is almost disappearing. Fourth, the crowd needs a direction, for example a crowd of the people walking on the pavement in the same direction. Individually everyone has their own goal destination, but overall the crowd has a common goal: to reach that destination [3].

Le Bon and McDougalls suggested that a crowd is not simply a combination of individual acts, but rather a social behavior being guided by forces defined by the collection (Collective Consciousness or Group Mind) [10][11][21].

Moscovicis relied on Le Bon's and McDougall's theories, to suggest that collectiveness relies on shared images and shared ideas to form the basis of common sense. These shared images and shared ideas become the cognitive context within the crowd, which members follow to collective communication and coordinate their actions [13]; [21]. These cognitive contexts are called Norms, and in the case of crowds are called Social Norms. Social Norms are rules of behavior that synchronize our communication with others.

A Crowd Behavior is carefully regulated by already existing social Norms. Those Norms are providing limits for the behavior but also members have to behave almost the same as their surrounding members in that crowd [6]. The group behavior will result to collective judgments and collective choice. As people behave almost the same with their neighbors, they adapt even more extreme judgments than they intended to [6]. Because of this behavior, most of the time, the decisions and preferences are conceptually distinct from the initial opinion of the individual [6].

3 Implementations

In order to take a step toward validation we implemented a prototypical scenario that exemplifies our model. We will describe crowd formation that has been triggered by curiosity. Example of a crowd stimulant is when an incidence, like an accident or human fainting or something that can causes people to be curious about, occurs in the middle of the road, or in the university. We divide the problem into two stages.

In the first stage of the particular crowd formation, an instance crowd formation is observed. Moreover, the first crowd formation is triggered firstly, by the need of people to help others, so the goal in this case is to help, or by the need to satisfy their curiosity if we accept that crowds form on daily basis, at moments that are very tragic and shocking [9]. People have the need to help their fellowman because it feels good, because the one that helps, he or she has something to give and by that have a feeling of being capable, "I can help, that means I am not worthless", and it is empowering, since it gives the helper a sense of control, [4]. The size of the this first group is depending on a lot of factors, like location of the accident, the appearance of the people that need the help, the culture and the abilities of the people that wants to offer help, and etc. Moreover, the size of the first crowd most of the times is very small. The collective behavior of these people that categorizing them as a crowd; is the willingness to help the person in need.

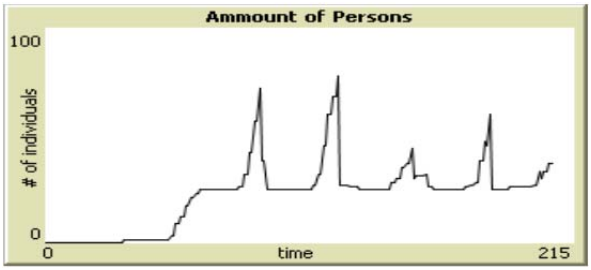


Figure 2. The density of people during the accident

The second reason that attracts the people up to that physical point is curiosity of the accident. Something, unexpected is changing in the picture of the normal world that people expected to live and to experience. That abnormal, and not dangerous for people, behavior that the particular person is presenting is sufficient to trigger the curiosity of the people that can have a visual with the incident, as it was explained earlier. Again, the size of this crowd is small, due to the reason that the people that experience the incident on the first hand are very limited. In the second stage of the crowd formation, crowd formation is being observed due to the necessity of the crowd to get bigger as Canetti mentions (Canetti, 1960). According to Canetti, the urge to crow is the first and supreme attribute of the crowd. People start streaming to that point, without even knowing what happened. A Canetti mentions that the movement of the people to that single point is being transmitted to others as well [3]. At this point someone can observe the collective behavior bonds (curiosity) that individuals are forming between them. The goal is to satisfy the curiosity and to understand the particular movement to a single reference point. At the second stage, the volume and the density of the crowd is much greater than in the first stage. Here crowd is formed in two waves, the first wave happens when people begin to arrive and try to join the rest of the crowd and the second stage is when the crowd renounces growth and puts a boundary to the number of people and the space that crowd is occupying. The second stage it can be compared to a vessel into which liquid is being

poured and the vessel is overflowed. The individuals that have satisfied their goal are leaving, and after a while are being replaced by others, see Figure 2.

4 Conclusion

The study of crowds is still at a very early stage. As the technology advances, we are going to be able to model and create agents that can mimic human beings and test the assumptions that researchers are making. Besides, with the help of the cameras, and Agent-based programs, we are able to create images that symbolize a basic pattern that a crowd creates. According to Keith Still [18], the patterns that each big crowd assumes, while it moves, are very identical and not random.

Acknowledgements

Author wishes to thank several students who contributed to discussions and implementations supporting this research, most notably among them P. Kamkarian, A. Chrsitau, H. Gajula, and M. Bouchard.

References

[1] F.H. Allport, 1933. *Institutional behavior*. Chapel Hill: University of North Carolina Press.

[2] D. Forsyth , 1990. Group dynamics. Belmond, Ca: Cengage Learning.

[3] E. Canetti, 1960. Crowds and power. New York: Continuum Publishing Corporation.

[4] Gartner, and F. Riessman, 1993. Making sure helping helps. Social Policy, 24(1), 35 – 36.

[5] M. Fox, 1997. Concepts in Ethology: Animal Behavior and Bioethics, Krieger Publishing Company.

[6] M. Hogg, 2000. Social Categorization, Depersonalization, and Group Behavior. In M Hogg (Ed.), Blackwell Handbook of Social Psychology: Group.

[7] E. Hutchins, 1995. *Cognition in the Wild*. Cambridge, MA.: MIT Press.

[8] M. McPherson, L. Smith-Lovin, and J. Cook, 2001. Birds of a Feather: Homophily in Social Networks. *Annual Review of Sociology*. 27:415-44.

[9] R. Lacks, J. Gordon, & C. Mccue, 2005. Who, what, and when: a descriptive examination of crowd formation, crowd behavior, and participation with law enforcement at homicide scenes in one city. American journal of criminal justice, 30(1), 01 - 20.

[10] Le Bon, C. 1896. The Crowd: a study of the popular mind. T. Fisher Unwin.

[11] W. McDougall, 1920. The Group Mind: A sketch of the principles of Collective Psychology with Some Attempt to apply them to the Interpretation of National Life and Character. Cambridge University Press.

[12] M. Minsky, 1988. Society of Mind, Simon & Schuster Pub.

[13] S. Moscovici, 1984. The phenomenon of social representations. In R. M. Farr & S. Moscovici (Eds.), Social representations (pp. 3 -69). Cambridge University Press.

[14] R. Moreland, 2006. Identification. *Proceedings of the Department of Homeland Security and Social Psychology Roundtable* .

[15] S. Reicher, 1996. The Battle of Westminster: Developing the Social Identity Model of Crowd Behavior in Order to Explain the Initiation and Development of Collective Conflict. European Journal of Social Psychology, *26,* 115-34, John Wiley and Sons.

[16] J. Scott, 1991. Social network analysis: a handbook . Sage Publications Ltd.

[17] H. Simon, 1955. A Behavioral model of rational choice. *The qurterly journal of economics*, *64*(98), 01-20.

[18] K. Still, 2000. Crowd Dynamics, Doctoral dissertation, University of Warwick.

[19] D. Sumpter, 2009. Collective Animal Behaviour. Princeton University Press.

[20] D. Taylor, & F. Moghaddam, 1994. *Theories of intergroup relations: international social psychological perspectives*. Westport, CT: Praeger Publishers.

[21] R. Tindale, H. Meisenhelder, A. Dykema-Engblade, M. Hogg, 2000. Shared Cognition in Small Groups. In M Hogg (Ed.), Blackwell Handbook of Social Psychology: Group processes (pp. 001-030). Blackwell Publisher Ltd.

Architecture for design and development of security systems based on agent technology

M. Guadalupe Cota[1], J. Pablo Soto[1]
[1]Departamento de Matemáticas, Ciencias de la Computación,
Universidad de Sonora, Hermosillo, Sonora, México.
{lcota, jpsoto}@gauss.mat.uson.mx

Abstract - *In this paper presents an hybrid architecture of agents for design and development of security systems, which is named ARSEC-AMS (Security Architecture for Multi-Agent System). ARSEC-AMS is organized in functional modules that facilitate the implementation of mechanisms for interaction and reasoning of agents. The architecture approach is different from other techniques and methodologies that have been applied in the development of protection tools, which have been successful but have been unable to control events on the topic "Day Zero", that occurs when new intrusion techniques that are used by malicious programs or users are not recognized by the protection programs and try to exploit vulnerabilities that compromise the computer resources. It is composed of the following modules: an interpreter to filter the events registered, a module of blackboard, a module deliberative-cognitive, a security module and a reactive module, which together are a platform for achieving the security goals previously established.*

Keywords: Architecture, Grammar, Agent, Security

1 Introduction

Agent technology is an innovative field in applications development, where agents are identified as software entities that can perform a series of operations with some degree of independence. They acquire forms of reasoning for perceive their skills and functional scope for decision making, using information from their environment and the knowledge provided by system designers [1] [2].

Although technological developments in computer sciences provides to all types of organizations and individuals such as the opportunity to upgrade their services and increase income and productivity, it is necessary consider security issues that can generate millions in losses on either information or money. Currently, most developed countries are making use of this technology and therefore are more susceptible to cyber crimes that aim to destroy or modify stored data or is transmitted for network, situation that forces them to adopt security measures designed to protect the operations performed by network and guarantee an optimal confidentiality level, integrity and information availability [3].

One of the most serious threats in computer security is called 'zero day', identified as the period where appears "malware" that not can be detected by security software and ends at the time that it may solve the problem. For instance, we have the case of the worms, which generally have the purpose of compromising resources of computer equipment that have been infected, exposing them to intrusions or attacks that exploit this situation until the problem is detected and controlled by security personnel, situation that may be last hours, months, years, or continue without a solution for an indefinite period of time [4] [5] [6].

After reviewing the specifications of the existing agent architectures as MESSAGE [7], BDI [8], GAIA [9] [10] and PRS [11], we conclude that none of them can be used individually to design security systems, and we decided to design a hybrid architecture for computer security area, which has been called ARSEC-AMS.

The remainder of this work is organized as follows. The next section we describe our proposal. In Section 3 the conclusions are presented.

2 ARSEC-AMS architecture

The ARSEC-AMS architecture (see Figure 1), is composed of modules that allow the agents interpret the events that occur in the environment, and to the designers implement mechanisms for reasoning deliberative-cognitive, criteria and levels of security, schedule actions of reactive type and maintain control of the blackboard which is used as an element of support in the communication system of agents.

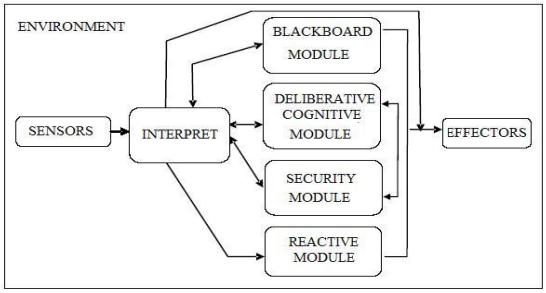


Fig. 1. ARSEC-AMS architecture.

2.1 Grammatical aspects

In this work it is suggested to use a different approach using ARSEC-AMS and grammar CLASS_W (Content-Language Windows Security System) [12] which is included in the 'content' field of structured messages in Agent Communication Language (ACL) [13] that is defined as standard by the Foundation for Intelligent Physical Agents (FIPA) [12]. CLASS-W is a Context Free Grammar (CFG) [14], and it will be denoted by G with G = (NT,T,P,σ), where

NT Not terminal symbols.
T Terminal symbols.
σ Initial symbol. ($\sigma \in NT$).
P Rules or grammatical productions where: $A \rightarrow a$ and $A \in NT$ and $a \in (NT \vee T)^+$.

In the following paragraphs, the syntax of CFG will be referenced in EBNF (Extended Backus-Naur Form) [14]:

The production (P) of initial symbol (σ) in grammar CLASS-W is:

<content>::= '(' < message > ')'
<message>::= '#'<key>'/' 'message''/' <transmitter_agent> '/' <type_agent> '/' <receiver_agent> '/' <type_conversacion> '/' <type_transaction> '/'

In CLASS-W '<type_conversacion>' is the symbol which indicates the type of transaction that is included in the message. The rest is for reference data of the agents and elements of the system working.

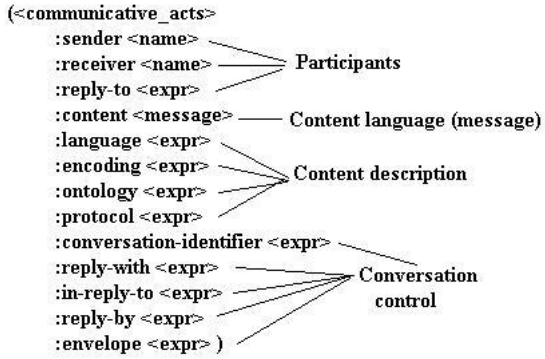


Fig. 2 Basic Format for ACL

The language of content that will be generated by CLASS-W is implemented on the label ***'content'*** within the framework of the basic format ACL (see Fig. 2), where communicative acts ***'sender'***, ***'receiver'*** and ***'content'*** are mandatory elements and the rest is optional.

2.2 Agent roles

The roles that have been designed to establish the different features of the types of agents have correspondence with a hierarchical level that defines the degree of authority, advantages and limitations that can assume the agents. The levels of hierarchy are associated with concepts related to the safety aspects such as the responsibility, permission and privilege.

In the hierarchy designed for the system of agents, the entities of Administrator Multiagent System (AMS) and Directory Facilitator (DF) are mandatory elements in the scheme of FIPA [15] and JADE [16].

Below is a description of the type of agents that can be created in ARSEC-AMS, in addition to the already mentioned from the prior paragraph:

a) The Agent AC is responsible for managing and coordinating the work of the agents and specialized perform and complex tasks.

b) The AL agent manages the internal workings of the local computer and is responsible for monitoring the implementation of security policies, review the behavior of the system resources, and monitor events in their environment.

c) The AR agent monitors the network performance from the local computer and performs monitoring activities related to network packet, detecting events that can be considered as abnormal and to implement actions previously established by the system administrator or an agent of type Coordinator.

2.3 Description of the functional behavior of agents

The behavior of agents associated with the modules of the architecture ARSEC-AMS can submit variations due to the identification of functions that correspond to the role and hierarchical level of the agent concerned, or by contextual changes that occur in their environment.

2.3.1 Interpreter module

The interpreter module is responsible to filter the events recorded in the system and follow up of the set of actions to achieve the objectives of multi-agent system and identify the semantics of the messages related to events filtered by the 'sensors'. The operation of this module is based on the flow of control that is shown in figure 3.

The basic tasks for AC agents are the following:
a) Obtain data related to the agents of the system.
b) Coordinate joint work.
c) Creating and controlling working sessions.
d) Manage instances of blackboard.
e) Implement guidelines and criteria established in the system and sent alerts to the administrator about

events that may be considered as security issues and implement preventative or corrective actions.

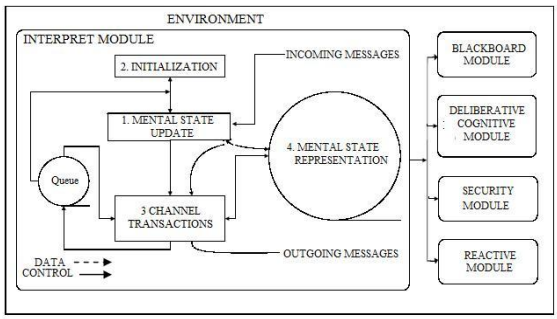


Fig. 3 Interpret module - Message Flow (AC).

Moreover, the AL agents are entities that provide basic support for the care, supervision and protection of local equipment, and together with AR agents must obey instructions given by AC agents.

Below describes the flow control of messages as shown in figure 2:

In points 1 and 2 is initialized and updated the mental state of the agents to identify beliefs, capabilities and scope of functions to each role.

In point 3 transactions and outgoing messages are controlled, the instructions are executed, and waiting list for transactions that must be rescheduled is controlled.

Point 4 is the basis of knowledge to represent the mind state of the agent (beliefs), which can be updated with collection of data related to the characteristics of agents, events, relevant information of incoming messages, plans (intentions), register targets or goals (desires), and state of the objectives registered.

2.3.2 Blackboard module

The blackboard architecture is a means of communication in a system of agents, a source of knowledge and a mechanism of control [17] [18]. In ARSEC-AMS, the blackboard is included as an element of recording the information generated in real time, which is a source of knowledge that is used in the process of analysis and identifying patterns that can be considered as potential attacks.

In the ARSEC-AMS architecture, the blackboard module is named as BLACKBOARD-ECRE (Blackboard-Control Element for Register of Events) which is shown in Figure 4 and has been designed to record significant events that can be analyzed for detect patterns that could be a security problem .

The grammar productions for blackboard module that are listed below corresponding to CLASS-W:

```
<blackboard>::=<registry_blackbrd>
    |<notify_blackbrd>|<end_blackbrd>
    |<wait_blackbrd>|<operation_blackbrd>
<registry_blackbrd>::='registry_blackbrd'
   '/'<agent> '/' <ontology_topics>
<notify_blackbrd>::='notify_blackbrd''/'
   <ontology_topics>
<end_blackbrd>::='end_blackbrd''/'
  <ontology_topics>
<operation_blackbrd>::='id_blackbrd''/'
   <namber> '/'<agent> '/'<ontology_f>
   '/'<inf_aditional>'/'<type_aport>
   |'id_blackbrd' '/'<namber>'/'<agent>
   '/'<ontology_f>'/'<type_aport>
```

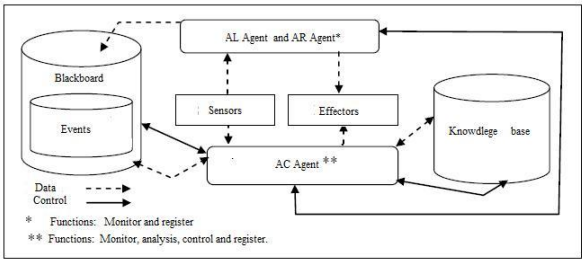


Fig. 4 BLACKBOARD-ECRE architecture.

2.3.3 Deliberative-cognitive module

An important aspect in the design of protection tools is to include components for record relevant information according to predetermined criteria, monitoring events, apply analytical techniques to classify and recognize patterns to identify potential threats and implement corrective or preventive procedures.

According to the above and considering that the agent-based systems can use forms of reasoning that enable them to interact and assume behaviors that define his level and functionality, the ARSEC-AMS architecture includes deliberative-cognitive module, through which can be implemented pattern recognition techniques, expert systems, logic-based rules, etc., and provides support in the mechanism of interaction for the decision making and knowledge discovery.

2.3.4 Security module

The security module works with the deliberative-cognitive module, allowing establish criteria and safety standards for the registration of warnings, instructions and actions as additional factors in making decisions. Below are some security transactions that are implemented through the grammar-W CLASS:

Instructions:

```
<instruction> ::= 'stop'<stopr>|'active'
  <active> |'review' <review> |'enable'
  <enable>|'disable'<disable>|'create'
```

```
<create>|'delete'<delete>|'modify'
<modify>|'update'<update>|'searchr'
<search>|'clasify' <clasify>
```

The instructions are related to actions that apply specifically on services, ports, applications, processes, policies, initial tasks, network settings, etc, which are defined as 'topics' in the corresponding database.

Alerts:

```
<alert>::='alert'\'/'<id_alert>\'/'<agent>
  \/'<level_security>\/'<ontology_topics>\/'
  <inf_ontology>\/'<action>
<inf_ontology>::= <list_processes>
  |<list_ports>|<list_services>
  |<list_task_initials>|<list_polices>
  |<list_changes_network>
<result_alert>::=<agent>\'/'<id_alert>
  \'/'<date>\'/'<time>
```

The alerts are controlled by coordinator agents.

2.3.5 Reactive module

Finally, in the case of reactive module can be programmed actions to respond to certain types of events previously set by the administrator or system designer, with the AL and AR agents who can assume this role.

A reactive agent is one that responds with actions planned to events that occur in real time. An example is the agent that monitors network packets by applying rules derived from a decision tree and detect network packets that may be harmful, or by applying pattern recognition techniques to identify text that is included in the network packet headers, the which are considered as a pattern that identifies certain types of malware. Therefore, this module is a fundamental part of the ARSEC-AMS architecture.

3 Practical examples

3.1 Deliberative-cognitive module

Whereas the logic programming is one of the methods most efficient and similar to what humans use to represent knowledge, the following is an example of a system of rules that can be applied in the deliberative-cognitive module, with scenario of 'ports' and the following assumptions:

a. We have detected that there is a ***worm scanworm***, which consists of '***client'*** and ***'server'***, the latter being that is downloaded to computers to infect and spread further to other computer equipment found in the radius about point where they are located.
b. The ***'client'*** is managed by the author of worm, and is dedicated to scan the network for computers that have opened the port '2127 ', which is opened in listening to the 'server ' waiting for connection 'client '.
c. Anomalies are found on infected computers caused by the 'worm', but there is no tool able for detect and eliminate to worm of the system.
d. It has issued an administrative warning and there is a rule in the logic module to close the port in connecting node, and prevent loss or theft of information until the problem is resolved.

The rule 'scanworm' is obtained from the following template:

```
semantic: scanworm(n_port ,t_operator,
        n_conexion, t_action).
        num_port(n_port).
        num_conexion(n_conexion).
        type_action(t_action).
rules:   scanworm(n_port, t_operator,
   n_conexion, t_action):-
        num_port(n_port),
        num_conexion(n_conexion),
        type_action(t_action).
datos: num_port('2127').
        num_conexion('0').
        type_action('close').
        type_action('review').
```

Interpretation: If the number port = '2127 'and number connections = '100', then it is 'scanworm' and action = 'close' and action = 'read '.

3.2 Blackboard module

Based on the scenario referred to in the above example, in blackboard module performs the following actions:

a. Search rules specified for the existing records on reported ports.
b. An example of a message CLASS-W in order to send instructions after the evaluation rule ***'scanworm***' and review records of blackboard is:

Messages sent for coordinator agent ('CAgent01') to agent ('Lagent10'):

"#0001/message/CAgent01/Coordinator/LAgent10/ private/instruction/close/pack_network/2127/"
"#0001/message/CAgent01/Coordinator/LAgent10/ private/instruction/review/pack_network/2127/"

Message sent for local agent ('LAgent10') to agent ('CAgent01'):

"#0001/message/CAgent01/Coordinator/LAgent10/ private/response/agree/"

The example based on references of virus and worms related with features that allow them to initiated automatically

when the operating system is loaded, using the registry where are tasks of system startup.

The situation described in this section relates to the plan that can be implemented when a 'unknown' process has been registered in the task start and is considered to be a potential security problem.

HKEY_LOCAL_MACHINE/software/microsoft/windows/currentversion/run/auto__hloader__key=c:\windows\system32\hloader_exe.exe
HKEY_CURREN_USER/software/microsoft/windows/currentversion/run/auto__hloader__key=c:\windows\system32\hloader_exe.exe

For purposes of illustration is used a current problem and some features of the Conficker Worm [19], which creates the files 'hloader_exe.exe' and 'hloader_dll.dll' in the system files folder ("c:\winnt", "c:\winnt\system32", "c:\windows\system"), inject this in the file 'explorer.exe' and recording with the following keys values:

Here is a scenario for the system of agents related to the topic 'startup_tasks', which can be implemented with CLASS-W:

a. Local agents have a registry of tasks authorized which will be loaded at system startup, and a list of processes that can run at any time.
b. Some local agents 'LAgent2' and 'LAgent10' have been found that the process 'hloader_exe.exe' is included as a startup task in the registry, and is not found on the list of authorized processes.
c. Local agents mentioned in the previous write on the blackboard, under the topic 'startup_tasks', its identification data and the process name 'hloader_exe.exe' and is classified as 'unkown task'
d. The coordinator agent 'CAgent01' revised the blackboard 'startup_tasks', and finds reported problem on 'hloader_exe.exe' and proceeds to do the following actions:

If there are rules associated with the process 'hloader_exe.exe' then the coordinator agent send instructions to the local agents, which reported the problem. If not, the following instructions are sent to the local agents:

1. Check if the process 'hloader_exe.exe' is active, and if so, disable it until it receives a new instruction to confirms or revoke the action.
2. Remove strings located in the registry in startup tasks that are related to the file 'hloader_exe.exe'.
3. Check that the 'hloader_exe.exe' process is blocked and that it is not in execution.
4. Record a message in the administrator container with the following information: topic 'startup_tasks', process name 'hloader_exe.exe' which has been regarded as unknown, and 'preventive' instructions were sent to local agents who reported the problem.

When the administrator reviews the container, a decision should be made whether to create a rule with the actions that need to be implemented or deactivated the precautionary measures, and for the process named 'hloader_exe.exe' will be registered the respective indication on the blackboard 'instructions'.

Later, when the coordinator agent reviews the blackboard 'instructions', turns the respective actions.

A plan like the above can help to avoid the problem 'zero day', particularly relating to the unknown process that has been registered as startup tasks and can be a potential threat.

4 Conclusions

The security in computer networks is a topic that is emerging day by day due to the massive data management, though it has increased the productivity and wealth of all kinds of organisms or businesses, it has also been the means by which have led to significant financial losses, since as the countries make use of these type of services, the goal becomes more vulnerable to attacks based on the information. To counter and control this problem, it is necessary to adopt safety policies, based on computer technology, that provide the possibility of setting safety standards that allow to increase the level of confidence in the exchange of information through this medium.

Although the vulnerability of the systems cannot be eliminated completely, and therefore, the timely detection of security problems plays an important role, the "zero-day threats" are what most worry security experts.

Among the various alternatives that exist to control this problem, are those based on Intrusion Detection Systems that allow monitoring events that might be associated with malicious or harmful actions, but have the disadvantage of generating a large number of alerts false, and which do not prevent attacks that arise when these are unknown to the databases used to support the application.

With a different approach to the current, this presentation introduces the proposal to use ARSEC-AMS architecture as a design tool to allow an expert, an administrator or developer of security systems, manage or develop system bases on agents and provide the possibility to harness and apply the experience acquired in area of security.

Finally, we should add that although there are different types of grammars for agents, there was no evidence of a grammar for design and development of agent systems to the field of security. The alternative of using CLASS-W based on ARSEC-AMS architecture which are presented in this work

can lead the way forward in the development of systems as we believe that agent technology has proved very efficient in other areas, and sometimes outperforms other techniques or methodologies which are not achieved the same results.

5 References

Number in square brackets (“[ ]”) should cite references to the literature in the main text. List the cited references in numerical order at the very end of your paper (under the heading `References'). Start each referenced paper on a new line (by its number in square brackets).

[1] J.M. Bradshaw, “Software Agents”, Boston: MIT Press. (http://agents.umbc.edu/introduction/01-Bradshaw.pdf), 1997.

[2] S. Russell & P. Norving, “Artificial Intelligence (A Modern Approach)”. Prentice-Hall, pp. 36-45, 57, 110, 151, 157, 161, 163, 165, 166, 171, 196-201, 304, 419, 1995.

[3] K. N. Randall, C. L. Pano, “Security for wireless communications (Telecommunications Series)”, McGraw-Hill, pp. xx, xxi, 14, 51-53, 73, 188-190, 195, 287, 2003.

[4] McAfee P. Security, “Microsoft Word 0-Day VulnerabilityIII”,http://vil.nai.com/vil/Content/v_vul27264.htm, 2006.

[5] R. Clark, “How to cope with security radical changes in cyberspace,http://www.symantec.com/region/mx/enterprisesecurity/content/expert/LAM_4198.html, 2006.

[6] D. P. Richard, “Viruses and Worms: What Can We Do About Them?”, Software Engineering Institute, Carnegie Mellon University, CERT® Coordination Center, Pittsburgh, Pennsylvania,USA,http://www.cert.org/congressional_testimony/Pethia-Testimony-9-10-2003/). 2003.

[7] G. Caire, F. Leal, P. Chainho, R. Evans, F. Garijo, J. Gomez-Sanz, J. Pavon, P. Kerney, J. Stark, P. Massonet, “MESSAGE - Methodology for Engineering Systems of Software Agents”. Technical Report Eurescom P907. http://www.eurescom.de/~public-webspace/P900-series/P907/index.htm, 2001.

[8] D. Kinny, M. Georgeff, A. Rao, “A Methodology and Modelling Technique for Systems of BDI Agents. Informe”. 1997.

[9] F. Zambonelli, N.R. Jennings, M. Wooldridge, “Developing multiagent systems: the Gaia Methodology”, ACM Trans on Software Engineering and Methodology, 12 (3). pp. 317-370, 2003.

[10] *GAIA*. http://www.auml.org/auml/documents/Gaia.doc.

[11] F. Ingrand, M.P. Georgeff. “Managing Deliberation and Reasoning in Real-Time AI Systems”, In Proceedings of the Workshop Innovative Approaches to Planning, Scheduling and Control, 284-291. 1990.

[12] G. Cota, P. Flores, J. Suárez, “CLASS-W, a Grammar for Security System Development based on Environment Equipment Agents with Windows-Technology NT”. Research in Computing Sciences. Advances in Computer Science and Engineering,pp.65-76, 2009.

[13] FIPA–Agent Communication Language Specifications. http://www.fipa.org/repository/aclspecs.html.

[14] Grune, Dick, Bal, E. Henri, Jacobs, J.H. Ceriel, Langendoen, G. Koen, Modern Compiler Design, pp. 548-596. McGraw-Hill/Interamericana de España, S.A.U, 2007

[15] FIPA – The Foundation for Intelligent Physicals Agents. (http://www.fipa.org).

[16] JADE–Java Agent Development Network. (http://jade.tilab.com/).

[17] M. Wooldridge & N.R. Jennings. Intelligent agents: Theory and practice. The Knowledge Engineering Review. Vol. 10 (2), pp. 116–152, 1995.

[18] N. Carver & V. Lesser., “The evolution of blackboard control architectures, Technical report”, University of Massachusetts Amherst, pp.19-23, 1995

[19] Conficker, http://conficker.com/.

Air Holding Problem Module to Decision Support in Air Traffic Flow Management

Leonardo L. B. V. Cruciol and Li Weigang
Department of Computer Science, University of Brasilia, Brasilia, DF, Brazil

Abstract— *Air Traffic Management (ATM) has the objective to guarantee that the aircraft operators meet the scheduled time of departure and arrival to maintain optimal flight profiles with minimum constraints. This paper describes a solution of Air Holding Problem (AHP) of ATM in Brazil using Multiagent System to improve Reinforcement Learning in collaboration with the flight controllers to support the decision process. The results obtained in the case study are promising, the behavior of the prototypes demonstrated that the Q-learning algorithm converged in a satisfying way. The prototypes generated actions that contributed effectively to the reduction of saturation in the air traffic scenarios under test.*

Keywords: Decision Support System, Multiagent System, Reinforcement Learning, Air Holding Problem, Air Traffic Flow Management.

1. Introduction

Brazil is preparing to organize two major world events: the World Cup of Football (2014) and the Olympics (2016). The air transportation growth experienced during the last decades has resulted to sharp airport congestion and Air Traffic Management (ATM) problems both experienced now and during the events. The Air Traffic Flow Management (ATFM) in tactical scenarios is determined to meet regulatory requirements in order to prevent the flight controllers to be submitted to a workload larger than that established by the present legislation. The researches have severe impacts on all airports' operational, logistic activities and Air Space Management (ASM) of the country.

Air Traffic Management has the objective to guarantee that the aircraft operators meet the scheduled time of departure and arrival to maintain optimal flight profiles with minimum constraints, while still following the safety standards. According to the literature review conducted in this research, the most recent papers discuss the use of various computer science formalisms, such as: Automata Theory [1], Intelligent Agents [2], Reinforcement Learning and Multiagent Theory [3,4].

Considering the complexity of Air Traffic Flow Control (ATFC) measures application, this paper presents a study conducted to develop a decision support device based on reinforcement learning that incorporates on it the Air Traffic Flow Management experience. The paper describes a solution of Air Holding Problem (AHP) applied to ATM in Brazil using Multiagent System to improve Reinforcement Learning of the software, in collaboration with the flight controllers.

Therefore, it is interesting to develop the software which can suggest more accurate and realistic actions, based on the past and on the expert's knowledge, to predict the impact of actions over the current local scenario and the next scenarios, both as locally as globally. At the same time, it is also important to increase the safety level of aircraft scenario because, when taking restrictive actions, flight controllers already have a prediction of their impacts.

The paper is organized in the following manner. In section 2, there is brief review of the Air Traffic Management concepts. Section 3 presents the methodology and application of the Multiagent System in modelling the Air Holding Problem. Section 4 contains the results of detailed case study of this research. Section 5 concludes the paper and proposes the direction of future research.

2. Present Scenario in ATFM

Air Traffic Flow Management (ATFM) focuses on the supply of information to maintain the traffic flow with safety and less impact on scenarios that are necessary to take unexpected measures [11]. The ATFM environment can be organized into three phases:

- Strategic Level: Consists on the tactical planning of flights covering the period of forty-eight hours until the time before the flight.
- Operational Level: Focuses on strategic decision making covering the period from forty-eight to two hours before the flight.
- Tactical Level: Consists on tactical decision making covering the period from two hours before the flight until the aircraft arrives at its destination.

This paper focuses on ATFM Tactical Level because it is the period which aircraft is in flight, which increases the problem level because the occurrence of the problem and its solution happens on real time.

It supports this decision making, once the high knowledge of flight controllers and Decision Support System (DSS) to improve is used, together, in this process. The ATFM is a very complex environment that every support to decrease impacts in the future on a determinate global scenario is important in this area.

2.1 ATFM in Brazil

ATFM in Brazil is going through a period of planning and adjustment to a high increase in the number of flights in the coming years. Only 44.94% of flights have their departure on time or advanced. The projection of air traffic in Brazil's sixty-seven airports studied is worrying if investments to expand capacity are not done [5,6].

The increase rates of flights are 1.00% to 1.88% at each airport. The number may seem small if looked only to the airport, but when this value is put in an environment, ATFM value grows proportionally with the number of airports included in Air Traffic Flow [7]. Taking into account also the time when these flights occur, it is possible to have peak intervals with a substantial number of aircraft, when compared to the current period.

Besides being prepared to manage the air traffic flow, the ATFM in Brazil needs to maximize the opportunities of the available structure so that resources can be put into good use. In this case, it can be mentioned the best management flow of the aircrafts while in the air and, therefore, besides improving the safety levels of the Brazilian aviation, it is possible to reduce negative indicators and achieve good results for all involved [5,8].

2.2 Air Holding Problem

AHP occurs in an ATFM environment when aircraft in flight route needs to wait on the air for a particular reason. One of the reasons could be that an airport had to be closed, for example. Situations involving natural phenomenon or terrorist acts, may be motive to excess of aircrafts on a particular sector and thus other aircrafts need to be retained for some time in other sectors.

The problem becomes much more serious, in critical scenarios, for example, when there is a scenario, which includes all aircraft under the responsibility of the Brazilian authorities and there are sub scenarios which responsibilities are under a certain Integrated Air Defense Center and Air Traffic Control (CINDACTA). When an action is chosen for a sub scenario, as the time goes by, the result of these actions can make better a sub scenario and aggravate another one, and then the whole situation can become an issue of great risk to the safety of everyone involved.

3. Decision Support System

Human beings and machines are complementary in several aspects. A well implemented decision support system is the one which helps other than take place of the human decision maker. The power of a taken decision by a human being in areas such as intuition, conceptualization and creativity are the weak points of a working machine. Human weakness, on the other hand, consists in aspects that a computer is accurate to achieve such as speed, parallelism, accuracy and the persistent storage of almost unlimited detailed information.

Therefore it is perceptible to realize a Decision Support System (DSS) combined between humans and computers means and abilities. The work made by automation should be only in areas like the control of problem solving activities, detection of conflicts and execution of evaluation, search and planning sequences. DSS enable to use data and models related to an entity of interest to solve semi-structured and unstructured problems with which they are faced [15].

The knowledge detained by some people is scattered, most times, including other people and systems, in this particular assignment sparse by other Flight Information Region (FIR), which is responsible for determinated sectors of air space.

The decision support systems allows working with problems of a decision making which proportion overtakes the normal rational capacity or exceeds temporal and financial means available. A flight controller specifies and models the decision process, by using a support to take decisions, so it can represent and manage the existing knowledge in the organization, control data increasing and finally take better decisions with precision.

3.1 Multiagent System

Multiagent Systems (MAS) are systems that can be described as multiple agents, which perform both autonomously and interact with other agents in the system. The agents take decisions on behalf of their goals and interact with other agents, in an independent form, to allow the performance of tasks to the satisfaction of a goal.

There are a variety of advantages in relation to a MAS in comparison to a system that consists of a single agent, which agent may indicate a lack of performance, reliability and easy maintenance to be subjected to a large number of tasks [12]. Besides, it is not possible a single agent provide specialized knowledge in ample amount.

The agents interact in a determinate environment, in which there may be one or more agents. In some cases, an agent can perform on his own, in others it can communicate with other agents and more advanced systems that may be necessary, there are responsible agents to intermediate this communication to control the environment. Multiagent Systems are divided into two main classes [13]:

- Open Systems: consists in agents that are not designed to achieve a common goal but participate in the process to reach the goal. Agents can enter and exit in the system at any time, in this case the entry of new agents must be considered.
- Problem Distributed Resolution System: agents involved are designed to reach a certain goal, in a cooperative and coordinated way.

A key element, in order to create a society of agents, is the ability to manage interactions and activities dependencies of different agents in a MAS

and coordinate them. Coordination is a key step in the MAS because these systems are inherently distributed.

3.2 Reinforcement Learning

Reinforcement Learning is a process which consists of an agent without prior baggage of knowledge and in a certain number of interactions, interacts with the environment to achieve its goals and therefore receives rewards for taken actions and achieved goals.

In this learning process is possible to realize an optimal solution to the problem in matter, in other words, it finds the best actions to achieve a determinate goal in the best possible way, taking into consideration a certain environment in an instant T_k, where k is the present instant.

The agent will act mutually with the environment, during the interaction, to realize which scenario it is and therefore it chooses the best action to be taken. After the agent presents an action, it will be given a value that is being called by reward. Therefore it could be possible check if taken action is better, or not, than actions taken in the past and it chooses the best for the current scenario [14].

4. Air Holding Problem Module

The Air Holding Problem Module (AHPM) use Multiagent System and Reinforcement Learning for the resolution of the problem. This module focuses on critical scenarios occurred in ATFM environment in Brazil, for which was made the actual data collect of the Air Traffic Flow Management scenario along with the Computational Model Laboratory for Air Transport (TransLab), Management Center of Air Navigation (CGNA) and Integrated Air Defense Center and Air Traffic Control I (CINDACTA I).

After the data collect, it was possible to propose a solution based on the received information for the problem. It is possible to model the scenarios including the current and the next after the application of restrictive action. To learn from the flight controller and improve its suggestions of scenarios over time and monitor, analyse and present information necessary for chosen solution.

AHPM is composed of four sub modules that interact with two databases. They will be described below, excepted for the Forecast Scenarios Module and Decision Support Module that will be explained in the next two subsections:

- Information Collection from Flight Information is responsible to search flight data, obtained from the CGNA, in a database and store information created by the flight controllers, which is, the maintenance and the use of the historical base.
- Reinforcement Learning Module is responsible for the software learning, which will receive the information from the Information Collection Module, calculate the reward of the actions and states in each scenario, given a particular instant in time. For that it is used the knowledge acquired by the agent in the past, stored in the learning database, achieving thereby a good result. These results are calculated by the algorithm Q-Learning, the best results are stored in a database to provide the necessary information in a next run.

The communication flow of AHPM is presented in Fig. 1 AHPM Overview.

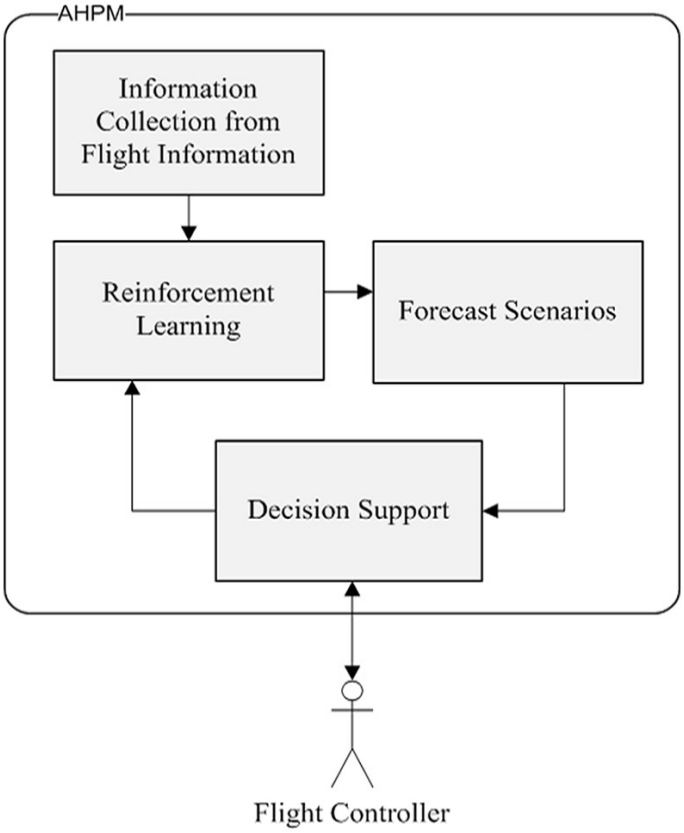


Fig. 1: AHPM Overview

4.1 Forecast Scenarios Module

Forecast Scenarios Module is responsible for setting the scenario in an instant T_{k+1}, in order to present for the flight controller, what will happen if the action suggested by the software is chosen, using graphs to display, at a later time, the scenarios impact by the chosen action [9,10]. This model makes the scenarios prediction and the processing, so that the best action for the scenario state is found. There are two kinds of agent, which are described below. The module to display forecast scenarios is presented in Fig. 2 Forecast Scenarios Module.

- FIR Control Distribution: It is responsible for collecting information from radars and performs the processing of the best action for the scenario of a particular FIR at a

given moment T_k. This agent can control other agents, if necessary, so it will have one more level. Some FIR's are more complex and work with more air traffic, so it is possible to subdivide some of them, if the action will improve their management.

- Central Control Distribution: It is responsible for pooling the best results achieved by the agent FIR Distribution Control and thus processes the best action seeking the lowest impact to the whole scenario involved.

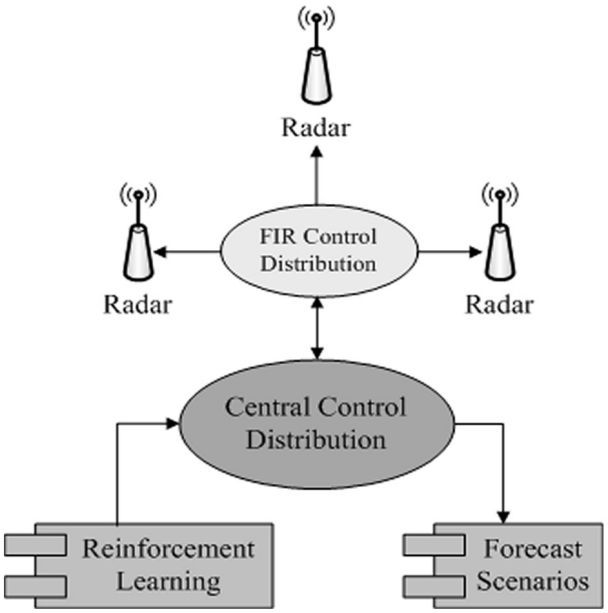


Fig. 2: Forecast Scenarios Module

4.2 Decision Support Module

Decision Support Module is responsible for integrating all work made in online manner. The module will present possible actions to be taken with each scenario that will be generated after the execution of certain action. It includes the impact of another scenario in the aeronautical map in the same state in a given instant of time. Therefore, an action performed in one state, in a given instant of time, needs to predict the outcome of its action in another scenario of air traffic flow management.

This module is the most complex because it is responsible to integrate collected information, display current scenarios and possible actions and scenarios states. The AHPM displays to the flight controller and waits an action to be taken. The module to interact with flight controller is presented in Fig. 3 Decision Support Module.

1) Current Scenario: It is responsible to display current scenario to flight controller;
2) Possible Solutions: It is responsible to display, based on Reinforcement Learning, possible local scenarios considering taken actions in the past;
3) Next After Action Scenario: It is responsible for presenting the hole picture in case the possible solution is chosen as the action;
4) Action Decide: It is responsible for receiving the action of the flight controller and send to the Learning Reinforcement Module for processing and storage.

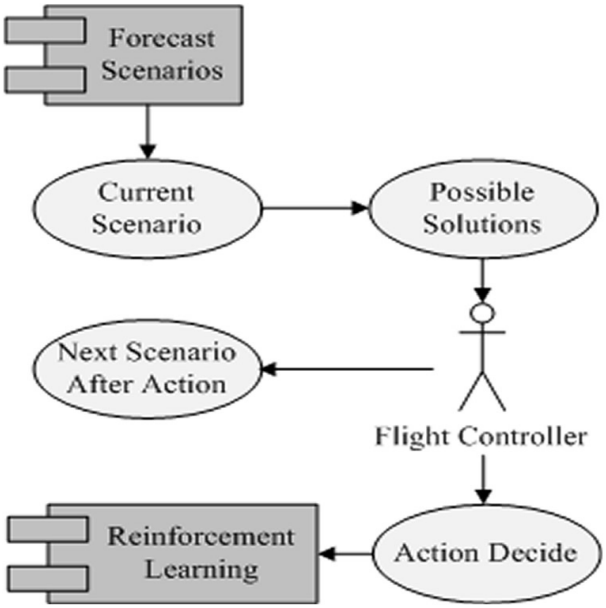


Fig. 3: Decision Support Module

4.2.1 Agent Modeling

The given model consists of two categories of agents: FIR Control Distribution Agent performs the computation needed to choose the best action for each FIR and Central Control Distribution Agent performs the computation needed, taking into account the best actions for each FIR and find the best action for the environment. An overview of the agent communication model involved in the proposed model is presented in Fig. 4 Agent Modeling.

4.2.2 Agent Learning

The agent learning is the process which the agent will learn, over time, to achieve the best actions according to the past. The FIR Control Distribution Agent will analyze each local scenario while Central Control Distribution Agent will analyze each results and choose the best action for future global scenario.

4.2.3 Reward Function

The reward of agent is defined by reward function that will present how much better it would be an action over

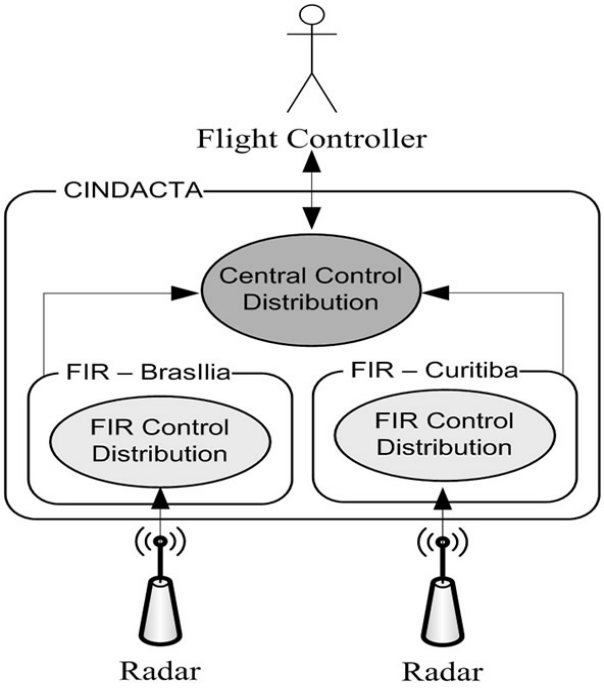


Fig. 4: Agent Modeling

others. The agent will take into consideration the capacity of aircraft by air space sector, it is set to twelve in Brazil, in other words, the action must reduce the amount of aircrafts and improve the flow of aircraft throughout the environment.

- FIR Control Distribution Agent: The objective of proposal reward function is to present an action that presents the least impact on the local current scenario. The reward function is defined by:

$$E_{FIR}(s) = \sum C_{FIR}(s) * F_{FIR}(s) \,|s \in S \qquad (1)$$

Where:

  - $E_{FIR}(s)$= it is the FIR scenario state.
  - $C_{FIR}(s)$= it is summation of aircraft in sectors s(congested) of the close joint sectors (S).
  - $F_{FIR}(s)$= it is summation of close sector from the sector s(congested).

- Central Control Distribution Agent: The objective of proposal reward function is to present an action that presents the least impact on the global current scenario, taking into account the actions suggested by the FIR Control Distribution Agent. The reward function is defined by:

$$H_{Central}(s) = D_{Central}(s) * G_{Central}(s) \,|s \in S \quad (2)$$

Where:

  - $H_{Central}(s)$= it is the environment scenario state.
  - $D_{Central}(s)$= it is summation of aircrafts in the environment.
  - $G_{Central}(s)$= it is summation of FIR's which sectors are congested.

5. AHPM Results

This research is being applied in Brazil ATFM environment and it is being applied, initially, for two FIR's: Brasilia and Curitiba. During tests in the AHPM, to validate this model, it was applied two reward function through Reinforcement Learning Module and it reached great results to reduce air traffic impact after restricted actions. The simulation of Brazilian Environment with two FIR's, Brasilia and Curitiba, is presented in Fig. 5 FIR - Brasilia and Curitiba Environment.

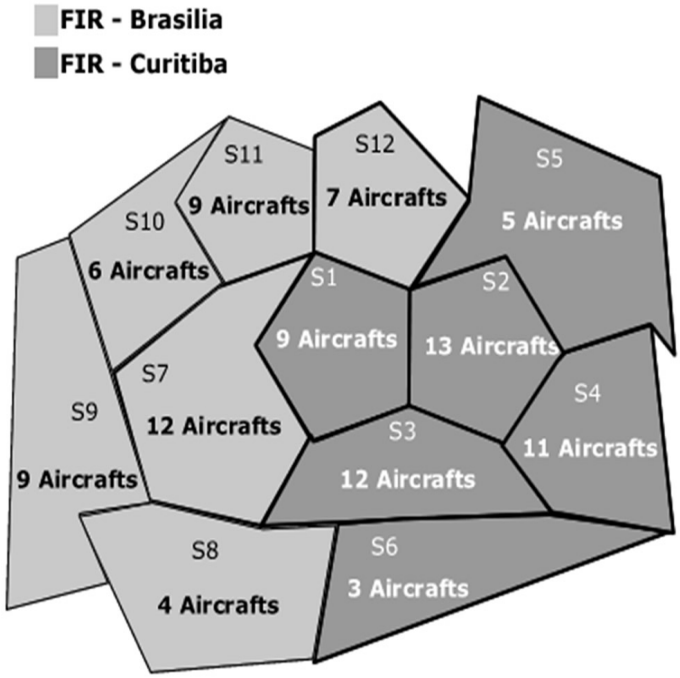


Fig. 5: FIR - Brasilia and Curitiba Environment

This research is to support the process of decision making, making possible for the flight controller to take the best forecast action of what will happen in the next scenarios for its FIR but always looking for the entire Brazilian environment. Then will be suggested to the flight controller always the best action for the global environment, therefore, taking into account all the others FIR's of the environment in instants T_{k+i}, where $k+i$ is a determinated future instant.

The FIR - Brasilia is the most important and critical in Brazil because it is responsible for more critical sectors than other FIR's in the country. The results in AHPM approach are presented in Fig. 6 AHPM Approach in FIR - Brasilia and Curitiba and Global Environment.

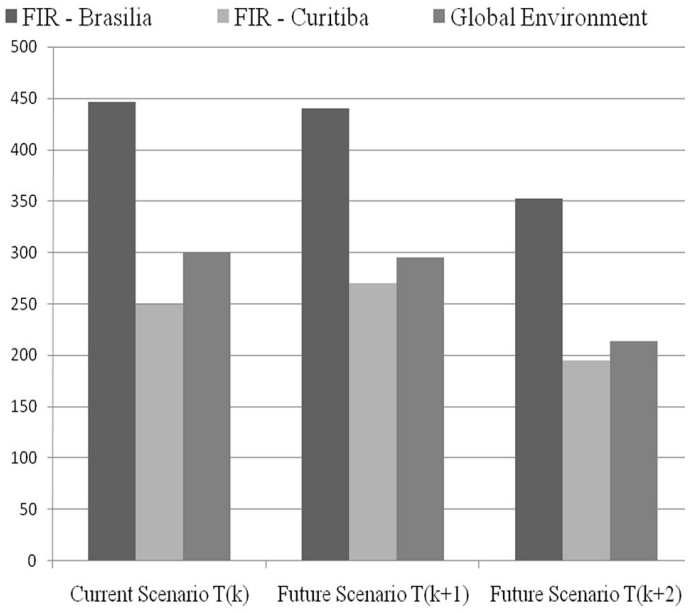


Fig. 6: AHPM Approach in FIR - Brasilia and Curitiba and Global Environment

Using reward functions within the module presented, AHPM, was possible to achieve the same result as reward for the global scenario, but with the use of the Forecast Scenarios Module it was possible to redistribute the congestion of aircraft by the sectors and reduce the impact of actions in the next scenario.

It was presented the results achieved with the distribution of congestion, taking the best action for the global environment and reducing the impacts of each action in the next instant in future. When the AHPM is used to forecast future scenarios improved approximately in 20% FIR - Brasilia, in 20% FIR - Curitiba and in 25% Global Environment.

6. Conclusions

The proposal model is developed to improve ATFM in Brazil, comprehending the Air Holding Problem. The research looks for an approach to the implementation of Multiagent System and Reinforcement Learning achieving innovative results in the area.

The presented decision support system is focused on some points, besides modelling the AHP, such as: reduction of Air Holding Traffic and impact of actions taken in the next local and global scenario, increase security and grounding for the flight controller to take decisions. The use of this proposal is unprecedented in the Brazilian environment and with the modelling of Air Holding Problem is possible to present an appropriate model to the reality in Brazil, together with air control, organs and research group specialized in this research area.

The research direction for future work is to construct computationally to implement this model in all Air Traffic Control in Brazil to incorporate in the life of the flight controllers.

References

[1] A. M. Bayen, P. Grieder, G. Meyer and C. J. Tomlin, "Lagrangian Delay Predictive Model for Sector-Based Air Traffic Flow", *AIAA Journal of Guidance, Control and Dynamics 28*, pp. 1015-1026, 2005.

[2] S. R. Wolf, "Supporting Air Traffic Flow Management with Agents", *American Association for Artificial Intelligence Spring Symposium: Interaction Challenges for Intelligent Assistants*, 2007.

[3] D.P. Alves, L. Weigang and B.B. Souza, "Reinforcement Learning to Support Metalevel Control in Air Traffic Management", In: Weber, C., Elshaw, M., Mayer, N. (Eds.), Reinforcement Learning - Theory and Applications, ARS, Vienna, 2008.

[4] A. Agogino and K. Tumer, "Regulating Air Traffic Flow with Coupled Agents Advances in Complex Systems", *Proceedings of 7th International Conference on Autonomous Agents and Multiagent Systems*, 2008.

[5] ANAC, "Justicativa da Portaria 676". (2009) [Online]. Available: http://www.anac.gov.br

[6] D.Q. Galvão, "O Caso Brasileiro: Usuários e Concessionário", *Seminário Internacional Sobre Concessão de Aeroportos*, 2008.

[7] L. F. Carvalho and C. J. P. Alves, "O Desafio da Infra-Estrutura Aeroportuária para a Copa do Mundo de 2014", *XV ENCITA*, ITA, 2009.

[8] T. H. L. Rolim, T. A. A. Portela and R. T. Almeida, "O Controle do Espaço Aéreo", ASCOM/DECEA, Technical Report, 2004.

[9] A. J. M. Castro and E. Oliveira, "Using Specialized Agents in a Distributed MAS to Solve Airline Operations Problems: a Case Study", *IAT*, 2007.

[10] A. Agogino and K. Tumer, "Distributed Agent-based Air Traffic Flow Management", *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, 2007.

[11] L. Weigang, C.J.P. Alves and N. Omar, "An Expert System for Air Traffic Flow Management", *Journal of Advanced Transportation*, 1997.

[12] G. Weiss, *Multiagent Systems, Artificial Societies and Simulated Organizations*, The MIT Press, 2005.

[13] M. Juchem and R.M. Bastos, "Engenharia de Sistemas Multiagentes: Uma Investigação Sobre o Estado da Arte", PUCRS, Technical Report, 2001.

[14] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*, The MIT Press, 1998.

[15] A.J.M. Beulens and J.A.E.E. Van Nunen, "The Use of Expert System Technology in DSS", *Decision Support Systems 4*, 421-431, 1988.

Challenges in Distributed Coalition Formation among Collaborative Multi-Agent Systems:

An Experimental Case Study on Small-World Networks

Predrag T. Tošić and **Naveen K. R. Ginne**
Department of Computer Science, University of Houston, Houston, Texas, USA
ptosic@uh.edu, ginne.naveen@gmail.com

Abstract - *We study collaborative multi-agent coalition formation in a fully decentralized, resource-bounded setting. In that context, we analyze, simulate and optimize* Maximal Clique based Distributed Coalition Formation (MCDCF) *algorithm. We continue our experimental, simulation-based investigation of MCDCF performance on types of underlying graphs that various research communities find important. In particular, we focus in this paper on MCDCF performance on the* small-world graphs. *We consider three classes of such graphs, where each class is defined by the probability of randomly rewired (that is, non-local) edges. We compare and contrast performances of the MCDCF protocol as the fraction of randomly re-wired edges is increased from zero (corresponding to k-regular rings) to 20% of the total number of edges in the graph. We discuss those aspects of coalition formation performance that turned out in accordance with our expectations, as well as those where the performance was not as expected. The main conclusions are twofold. One, the exact structure of sparse-on-average graphs can make a great deal of difference when it comes to the scalability and efficiency of our graph partitioning based approach to coalition formation. Concretely, the algorithm performs much better on Erdos-Renyi random graphs with modest average node degrees and several hundreds of nodes than on small-world graphs of the same or comparable densities and sizes. Two, some of the candidate coalition search mechanisms in MCDCF need to be improved in order to ensure better overall performance when our algorithm is run on small-world-like topologies. We hope to report improvements in that regard in our near-future work.*

Keywords: distributed algorithms, distributed AI, multi-agent systems, coalition formation, small-world networks

1 Introduction and motivation

We study, devise and analyze scalable distributed graph algorithms that are useful in addressing challenging and important coordination problems in *Distributed Artificial Intelligence* (DAI). We are interested in artificial agents such as softbots, robots, unmanned vehicles or smart sensors that are fully autonomous, capable of communicating with each other, and therefore potentially able to self-organize in various ways in order to accomplish tasks that would exceed the computational, sensing and/or acting capabilities of the individual agents. Of our particular interest are mechanisms and protocols that enable large ensembles of such autonomous agents to coordinate with each other in a *fully decentralized* manner, and collaborate in order to accomplish various tasks. The kinds of agents we have in mind are referred to as *distributed problem solvers* (or DPS agents) in the DAI literature [2, 22, 23].

Among various interesting problems in distributed coordination and control of such ensembles of DPS agents, we have been extensively studying genuinely autonomous, fully distributed, dynamic multi-agent *coalition formation* [15 – 18, 20, 21]. We have approached coalition formation from two perspectives: one is the *multi-agent coordination* view of coalition formation, and the other is the *distributed graph algorithm design* view. We have integrated these two views and designed an original, fully distributed, scalable and robust coalition formation graph algorithm appropriately (if somewhat awkwardly) named, *Maximal Clique based Distributed Coalition Formation* (MCDCF) [15]. In this paper, we build on the top of our earlier work on MCDCF [15-17, 19-21], summarize and interpret our recent simulation results on small-world underlying networks, and outline our ongoing and near-future research on further improvements and scaling-up of our algorithm.

The rest of this paper is organized as follows. In Section 2, we motivate the problem of multi-agent coalition formation, briefly review some approaches to this problem found in the literature, and outline the problem setting that we have been studying in our prior and ongoing work. In Section 3, we summarize our MCDCF algorithm for coalition formation among collaborative agents. In Section 4, we present, analyze and discuss our recent simulation results when MCDCF is run on three types of small-world graphs. Section 5 summarizes the paper and indicates some of the major challenges ahead.

2 Distributed coalition formation

Distributed coalition formation in multi-agent domains is an important coordination and collaboration problem that has been extensively studied by the Multi-Agent Systems (MAS) research community (e.g., [1, 2, 8-10, 12, 15-21]). There are many important collaborative MAS applications where autonomous agents need to form groups, teams or coalitions. Reasons behind multi-agent coalition formation can vary. In the context of *Distributed Problem Solving* (DPS) collaborative agents, possible motivations include, but are not limited to, the following: to share resources, to jointly

complete tasks that exceed the abilities of individual agents, and/or to improve some system-wide performance metric such as the speed of joint task completion [7, 12, 17].

One well-studied general coalition formation domain is a collaborative multi-agent environment populated with a variety of distinct mutually independent tasks, where each task requires a tuple of resources on the agents' part in order for the agents to be able to complete that task [10-12, 17, 18]. In this *distributed task allocation* context, agents need to form coalitions such that each coalition has sufficient cumulative resources or capabilities across the coalition members in order to be able to complete the assigned task.

While *distributed problem solving* and, in particular, *distributed task or resource allocation* are perhaps the most common MAS contexts in which coalition formation has been studied, the problem of designing and analyzing protocols for efficient coalition formation among autonomous agents has arisen in other contexts, as well. In the most general setting, the conceptual and mathematical foundations of the coalition formation process have been addressed from the standpoint of *algorithmic game theory* and *mechanism design* [22, 23]. Game-theoretic models and techniques have been considered particularly valuable when the agents that engage in forming coalitions belong to different designers and/or organizations; such agents in general tend to be *self-interested* (as opposed to *altruistic*).

Among other reasons why agents may engage in cooperative coalition formation, limitations on sensing, computational (processing and/or memory), energy (e.g., batter life in wireless sensors; or fuel tank capacity in UAVs), communication and acting resources of individual agents have been prominent in much of the prior research on various types of multi-agent encounters [8, 23]. A thorough survey of the state of the art (covering most of the important directions of coalition formation research up to approximately year 2005) can be found in [7].

We study the problem of distributed coalition formation in the following problem setting. We assume a collaborative multi-agent, multi-task dynamic and partially observable environment. The tasks are assumed *mutually independent* of each other. In general, different tasks may have different values or utilities associated with them; moreover, the utility of a particular task may be differently perceived by different agents. This problem setting is particularly appropriate for many MAS applications involving *team robotics* and *autonomous unmanned vehicles* [5, 17, 18]. In particular, our agents are distributed problem solvers: they are assumed to be *strictly collaborative*, not selfish. The agents have certain capabilities that (may) enable them to service various tasks. Similarly, the tasks have certain resource or capability requirements, so that no agent or coalition of agents whose joint capabilities do not meet a particular task's resource requirements can serve that task [10, 16, 17]. Each task is of a certain *value* to an agent. Agents are assumed capable of communicating, negotiating and making agreements with each other [2, 8, 10, 17, 22]. Communication is accomplished via exchanging messages. This communication, however, is not free, which is why it is desirable to limit the amount of communication per agent [17]. An important difference between our problem setting and that in [10] is that we assume that *an agent's resources are not transferable to other agents* [15-17]. Thus, the only way for an agent A_i to use the internal resources of agent A_j for the purpose of servicing some task is that A_i and A_j join the same coalition, and then jointly attack that task.

Our *distributed maximal clique-based coalition formation* algorithm is based on the idea that, in a *peer-to-peer* MAS, an agent would prefer to form a coalition with those agents that it can communicate with directly, and, moreover, where every member of such potential coalition can communicate with any other member *directly*. That is, the preferable coalitions are (maximal) cliques [15-17]. Finding a maximal clique in an arbitrary graph is **NP**-hard in the centralized setting [3, 4]. This implies the computational hardness that, in general, each agent faces when trying to determine the maximal clique(s) it belongs to. However, if the degree of a node is sufficiently small, then finding some or even all maximal cliques this node belongs to may be feasible. If one cannot guarantee that all the nodes in a given underlying network topology are of small degrees, then the system designer and/or deployment coordinator may want to impose additional constraints, in order to ensure that the agents' coalition formation according to the MCDCF protocol is actually computationally feasible [15-17]. Alternatively, the system designer may want to consider modifying the objective function, so that the candidate coalitions an agent would consider forming need not necessarily be *maximal cliques* that this agent belongs to. The most appropriate design choice clearly depends on the application at hand, the nature of agents' tasks, and the agents' communication and computational resources [17].

3 The MCDCF algorithm

The MCDCF coalition formation protocol [15-18] is a *distributed graph algorithm*. The abbreviation stands for *Maximal Clique based Distributed Coalition Formation*, and was originally introduced in [15]. The underlying undirected graph captures the communication network topology among the agents. Each agent is a node in the graph. The necessary requirement for an edge between two nodes to exist is that the two nodes be able to directly communicate with one another. (Note: while communication links in practice are directional, the MCDCF algorithm works for both directed and undirected underlying graphs.) Communication takes place by local or *group broadcasts*. The group broadcast nature of the assumed communication model is primarily inspired by the applications that drove the original MCDCF design, namely team robotics and autonomous unmanned vehicles, in particular *micro-UAVs* [18, 19].

The basic idea behind MCDCF is to *partition* the underlying graph into (preferably, maximal) *cliques* of nodes. These maximal cliques would in practice usually also need to satisfy some additional criteria in order to form coalitions of desired quality. These coalitions are then maintained until they are no longer preferred by the agents - that is, when they are no longer sufficiently useful or even meaningful.

To ensure overall scalability and computational cost feasibility of the protocol, the candidate coalitions in MCDCF are required to be cliques of uniformly bounded sizes. This requirement, in practice, means one of the two possibilities. One possibility is that it is *a priori* known that the underlying MAS network topology is such that it can be guaranteed that there are no cliques that are prohibitively large. Otherwise, the system designer, based on the application at hand and the available system resources, a priori chooses a threshold $K = K(n)$, where n is the total number of nodes, such that only the coalitions of sizes up to K are considered.

Agents form coalitions in a fully distributed manner as follows. Each agent (i) first learns of who are its neighbors, then (ii) determines the appropriate candidate coalitions, that the agent hopes are (preferably maximal, but certainly of sizes bounded by K) cliques that it belongs to, then (iii) evaluates the utility value of each such candidate coalition, measured in terms of the joint resources of all the potential coalition members, then (iv) chooses the most desirable candidate coalition, and, finally, (v) sends this choice to all its neighbors. An agent also receives similar coalition proposals from some subset of its one-hop neighbors. The agent compares its current coalition proposal against those received from the neighbors. If there is an agreement among all agents in the current proposal of our agent, let's call it A, then this condition is recognized and a new coalition is formed. Once this happens, A's other neighbors, namely, those not in the newly formed coalition (if there are any such neighbors left), are notified of the formation of the new coalition. If there is no agreement, meaning, if at least one agent in the current proposal of A does not agree with that proposal, then the process continues and the protocol moves to the next round.

The basic procedure outlined above is repeated, together with all agents updating their knowledge of (a) what are the preferred coalitions of their neighbors, and (b) what coalitions have already been formed – until eventually every agent has joined some coalition. Note that, when an agent has no available coalition partners left, it necessarily forms the trivial, "singleton coalition"; hence, it is guaranteed that each agent will join some coalition after finitely many rounds (and it may possibly end up being the trivial coalition).

Once an agent has joined a coalition, that agent is done with the MCDCF execution. Once every agent has joined some coalition, the entire coalition formation process terminates. The round in which the last agent has joined some coalition is the final round; the total number of rounds until termination is our main criterion of our algorithm's computational efficiency and feasibility. We have shown in details elsewhere that, assuming a sufficiently sparse underlying network topology and under some additional, relatively mild assumptions that typically hold in application domains of our main interest, MCDCF is always guaranteed to converge after finitely many rounds [15-18].

A detailed description and pseudo-code for the algorithm can be found in [17]. Recent improvements and optimizations are summarized in [20]. These optimizations have been tested on structure-free random graphs and have generally produced very encouraging results [21]. One of the goals of the present study is to determine, just how effective these optimizations are when MCDCF is applied to *small-world-like* underlying topologies.

MCDCF is characterized by (i) being *fully decentralized* and genuinely *peer-to-peer* (P2P) and (ii) requiring only a very *coarse-level synchrony*. Property (i) is immediate: there is no central control of any kind, and no agent has any special role. In fact, all agents execute the same code. Moreover, MCDCF is a *highly local algorithm*, in a sense that there is no *network flooding* (or any other similar method of global information sharing). Due to property (i), and unlike many other multi-agent coalition formation protocols and other coordination techniques found in the MAS literature, MCDCF is rather robust and has no single point of failure. In particular, it works as intended in the presence of possible multiple node and/or communication link failures. However, these node and link failures have to be of a "nice" enough kind ; more details about the robustness and fault tolerance properties of MCDCF protocol can be found in [17].

By *coarse synchrony* we mean that no agent begins the next round before all its neighbors are done with the previous round. In particular, we assume that each agent uses a timer so that, once no response from a particular neighbor is received within the allocated time slot, that particular neighbor is removed from the agent's extended neighborhood list and, therefore, also from the current (and all future) candidate coalitions for that agent. However, within a given round, different agents may be executing different stages of that round; again, we refer the reader to [15–17] for more details.

4 MCDCF on small-world graphs

We simulate the MCDF algorithm using Java. We have conducted experiments to investigate how well MCDCF performs on small-world graphs, and also to verify general theoretical predictions related to scalability [16, 17]. The two most important parameters for the MCDF algorithm are the number of nodes n and the average node degree avg_deg. To experimentally validate (or challenge) the performance predictions that we made in our prior work based on graph-theoretic considerations [15-17], we considered two basic scenarios. In the first scenario, we maintained the number of nodes as a constant and varied the average node degree. In the second scenario, the average node degree is constant and we varied the total number of nodes.

The assumptions we made throughout the simulation experiments are as follows:

1. We have used the Watts-Strogatz model [25] to generate graphs with *Small World* structures. The graphs have been generated using the software *NetworkX*, a Python package to generate complex networks [24].
2. We have generated the small-world graphs starting with a regular k-ring and then, with probabilities 0, 0.1, and 0.2, respectively, we would rewire randomly selected local edges (in the original regular ring) and replace those selected edges with random (hence, in general, non-local) edges.

3. To make it an apple-to-apple comparison with our experimentation with Erdos-Renyi random graphs [21], we used the edge replacement operation as described in remark 2 so that we control the exact total number of edges in the graph, and hence the exact average node degree.
4. The number of rounds until convergence and the total number of coalitions in the final coalition structure are both computed as (arithmetic) means over 10 simulation runs where a different graph is generated randomly at the beginning of each run.
5. The maximum coalition size (see Figure 2 below) is the maximum size among all coalitions found across all of those 10 simulation runs (for a given pair of parameter values (*n, avg_deg*).

Scenario 1:

The varying parameter in this scenario is the average node degree and the number of nodes is held constant (n = 200). We have incremented the average node degree in multiples of two, from *avg_deg* = 4 up to *avg_deg* = 20. We summarize the main lessons learned from the simulation results for this scenario and their statistical analysis. In each plot in this subsection, the x-axis captures the average node degree of the underlying graph. Specifically, the average node degree plotted along the x-axis ranges from 4 to 20 in the increments of 2.

The plot capturing the dependence of the number of rounds on the average node degree shows a rapid growth rate when the probability of random rewiring is greater than zero (Figure 1). The MCDF algorithm has been argued in our earlier work to scale well for sparse underlying graphs, and our simulation results for Erdos-Renyi random graphs with increasing densities validate our prior theoretical predictions in [16,17]; see [21]. However, for small-world graphs with non-zero probability of randomly rewired edges, the number of rounds, at least in the range of graph densities captured in our simulations, appears to grow exponentially. This is rather alarming; one implication is that better mechanisms for how MCDCF traverses the lattice of candidate coalitions of a given node (see [20] for details) need to be sought.

We recall that, as the average node degree increases, so does the average size of a node's neighborhood list, and hence the number of candidate coalitions to consider. Moreover, the number of candidate coalitions grows *exponentially* with the number of neighbors. The mechanisms for optimizing this search of the lattice of candidate coalitions that each agent (node) locally performs are discussed in detail in [20]. While these mechanisms and optimizations have resulted in very good scaling and convergence behavior for the Erdos-Renyi graphs [21], the results for small-world graphs with non-zero probabilities of random re-wiring indicate poor scalability. That is, for the small-world graphs, the current mechanisms in MCDCF that are supposed to ensure fairly fast convergence in practice (as long as the underlying graphs are relatively sparse) may be inadequate.

For the random rewiring probabilities of 0.1 and 0.2 the convergence is much slower compared to when the rewiring probability is 0. The reason behind this is the graph generated using the Watts-Strogatz model with probability 0 is a k-regular ring with n nodes, where each node is connected to its 2k nearest neighbors [24]. This property of near-by nodes having multiple common neighbors, according to our intuition, should enable the agents to (i) find and agree to non-trivial, relatively sizable maximal cliques, and (ii) be able to do so relatively fast. These predictions have been partly validated, as can be seen from Figures 1 and 3. We also found the following interesting property of MCDCF's behavior on regular k-rings: when the probability of random rewiring is 0, we typically get three different kinds of coalitions: one relatively large coalition of the size k + 1 (where k is the number of nearest neighbors on each side of a given node, for the total of 2k), one doubleton, and the rest of the nodes end up as singleton (i.e., trivial) coalitions.

As we increase the probability of rewiring in the Watts-Strogatz model the likelihood of nodes having multiple common neighbors, in general, decreases; consequently, we expected to see both the number of rounds to form coalitions to increase, and the coalition structure quality (cf. measured by average coalition size at the termination of the algorithm) to deteriorate. These predictions have been validated; however, the slow-down in terms of what appears as an exponential increase in the number of rounds until convergence is quite alarming, as further discussed below.

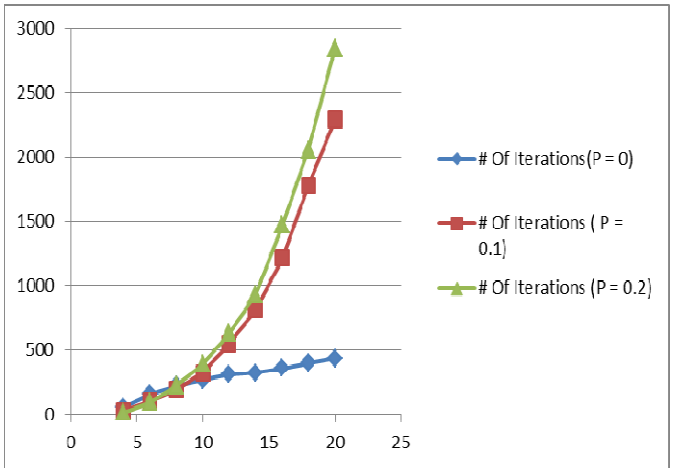


Figure 1. *Rates of convergence for three values of probability of random re-wiring in small world graphs, as a function of graph density; the y-axis is the number of rounds until convergence.*

The plot in Figure 2 captures the dependence between the average node degree and the maximum coalition size. As the density of the graph increases, the number of candidate coalitions per node also increases. We have observed in simulations that the nodes initially have many candidate coalitions that are bigger in size as the graph density is increased. However, the coalitions the agents have proposed in the earlier rounds need not be agreed by all the neighbors to whom those coalition proposals were sent. As the negotiation on coalition proposals continues and the agents fail to reach agreements round after round, those agents that have failed to reach an agreement eventually get to drop these relatively larger candidate coalitions that have been proposed but not

agreed upon; the agents then end up having to select candidate coalitions that are smaller in size. (That is, each agent's search for a candidate coalition is monotonic across the lattice of possible coalitions with respect to the usual subset relation; see [17, 20] for details). Therefore, higher average node degrees need not imply that larger-sized coalitions will necessarily be formed. However, we admittedly still fall short of complete understanding of the experimentally observed non-monotonicity in the maximum coalition sizes as the function of the average node degree (Figure 2); further investigating this phenomenon is subject of our ongoing work.

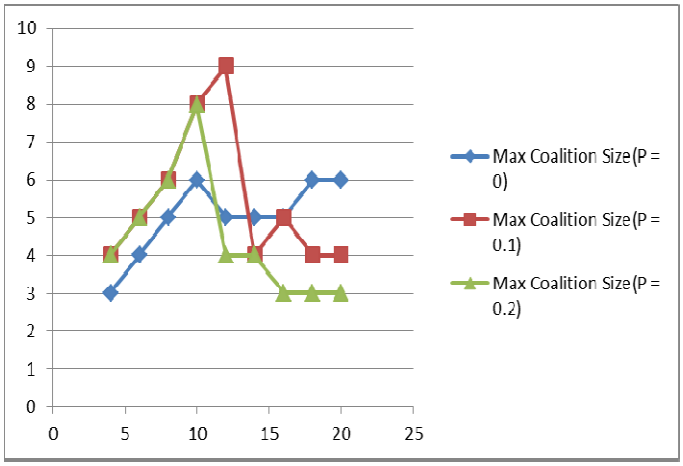


Figure 2. *Maximum coalition size as a function of graph density. The y-axis is the maximum size of an obtained coalition among all coalitions across all the runs with the same parameters values.*

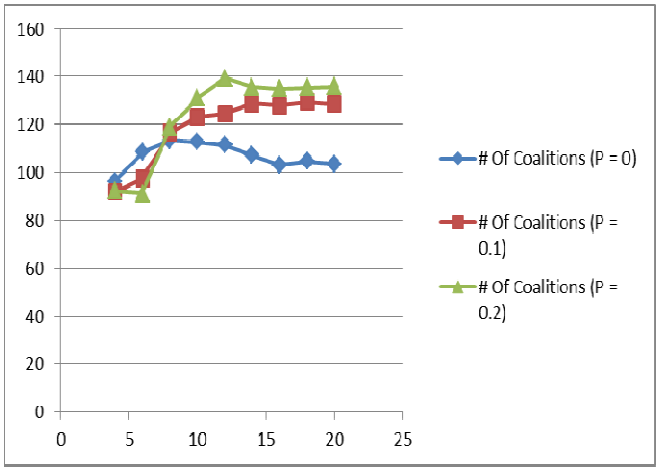


Figure 3*. The number of formed coalitions as a function of graph density. The y-axis is the average total number of coalitions formed across ten graphs with the specified parameter values.*

The plot showing the dependence of the total number of coalitions formed as a function of the average node degree also shows non-monotonic behavior (Figure 3). We observe that, even with higher average node degrees, a vast majority of nodes end up in either trivial singleton coalitions or else doubletons. In particular, coalitions of size 3 or greater are few and far between. We therefore find the quality of the resulting coalition structures, in general, to be below our expectations for this class of underlying graphs.

This behavior may be an indication of potentially serious issues with how MCDCF handles small-world graphs: as discussed earlier, its mechanisms for optimizing search of the lattice of candidate coalitions and the tie-breaking mechanisms (when a node has two or more *equally preferable* candidate coalitions to choose from) appear to work very well for the *structure-free* random (Erdos-Renyi) graphs [21], but the MCDCF performance on non-trivial small-world graphs of the same sizes and densities as those in [21] is much worse. Therefore, we need to further investigate new optimization mechanisms for MCDCF that would enable our protocol to better handle the small-world graphs.

Scenario 2:

In this scenario we vary the number of nodes and the average node degree remains constant: *avg_node* = 10 throughout. We increment the number of nodes by 20, starting from n = 40 up to n = 200 nodes. As in *Scenario 1*, for each pair of values (n, *avg_deg*), we run MCDCF on ten different, randomly generated graphs. Just like in *Scenario 1*, the captured values for the metrics of interest in Figures 4 – 5 are the actual arithmetic averages across all ten runs for a given pair of parameter values n and *avg_deg*. We do not show the plot for the maximum obtained coalition size as a function of the total number of nodes (i.e., the analog of Figure 2 in the previous subsection), as we did not find any systematic dependence or correlation between those two problem parameters. We next share several important observations and what we consider to be the main lessons learned from this scenario.

In Figure 4, we see that the number of rounds to reach convergence increases approximately *linearly* with the total number of nodes. According to theoretical predictions [16, 17], in general, how long it takes for MCDCF to converge primarily depends on the neighborhood sizes, not the total number of nodes in the graph. More specifically, how long will a typical node take to agree with some subset of its neighbors on which coalition to form, is cf. determined by the *neighborhood size* and hence (i) the size of the initial candidate coalition and (ii) the total number of candidate coalitions that may be considered during the protocol's execution. However, we recall that the total number of rounds is, in essence, determined by the slowest agent to form a coalition. Therefore, it is not surprising that as the number of agents increases, the convergence slows down (at a sensible rate) even though the average neighborhood sizes are held fixed: more nodes overall likely in practice results in more "slow" nodes, hence slower convergence. This increase in the average number of rounds with an increase in n is only slightly more pronounced when some random edges are added. In particular, even for regular k-rings, the convergence slows down linearly with an increase in the total number of nodes. As the probability of random edge re-wiring is increased, the linear slow-down with increase in n is still observed, but with slightly greater slopes. While not

necessarily surprising, we certainly find this experimental insight interesting.

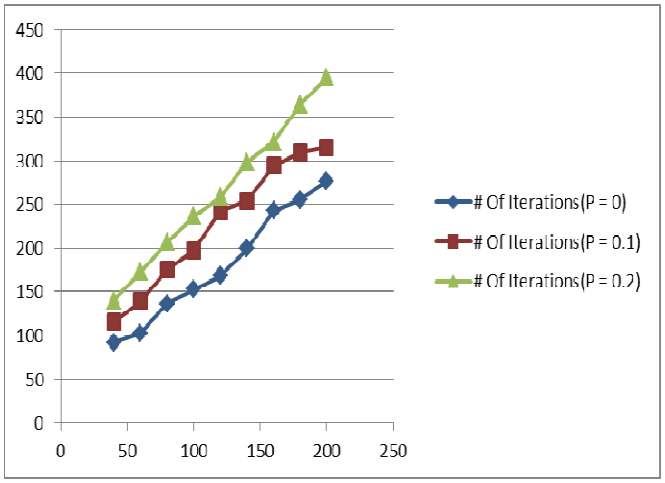


Figure 4. *Convergence rates for the three studied values of probability of random rewiring as a function of the total number of nodes. The y-axis is the average number of rounds until convergence.*

The plot in Figure 5 captures the relationship between the number of nodes and the total number of coalitions formed; the number of coalitions is a *monotonically increasing* function of n as well as of the probability of rewiring, as we expected. Moreover, the dependence on the number of nodes appears to be linear. This is in general accordance with our predictions, since the *fixed* average node degree implies that we shouldn't expect more larger-sized coalitions as n increases; rather, we get more coalitions of roughly the same average sizes as for smaller n. Therefore, this is one aspect of MCDCF performance where the theoretical predictions, our conceptual intuition and our subsequent simulation experimentation are all in agreement.

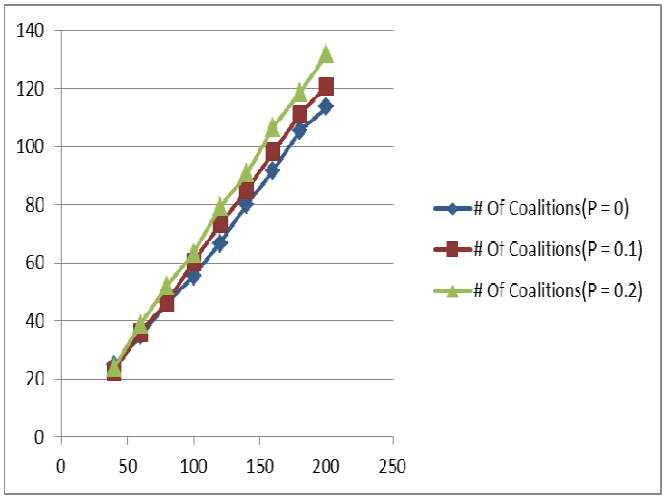


Figure 5: *Number of coalitions formed as a function of the total number of nodes. The y-axis is the average number of coalitions formed for the randomly generated small-world graphs with the specified number of nodes.*

We also note that, for rewiring probabilities of 0.1 and 0.2, due to random nature of our test graphs, most nodes end up in singleton and doubleton coalitions basically irrespectively of n and *avg_deg* (see also Figure 3). Therefore, in particular, as the number of nodes and the probability of rewiring increase, the total number of formed coalitions in general also tends to increase.

We recall that, in the original MCDCF algorithm, a node selects the *lexicographically first* coalition among the two or more equal candidate coalitions, as the tie breaking mechanism [16, 17]. This method works fine when the underlying graph does not have any particular structure – for instance, for random (Erdos-Renyi) graphs [21]. In the case of the Watts-Strogatz small-world graph model [24], when the probability of rewiring is 0, then the graph is a regular *k-ring* (k neighbors to the left, k neighbors to the right) with n nodes, where every node forms edges with its k nearest neighbors on each side. As we increase the probability of rewiring, the graph becomes more "random", but there's still structure to it – namely, the small world structure; in theory, this small-world property holds as long as the probability of rewiring is strictly less than 1 [25].

Lexicographically selecting one among several equally good candidate coalitions, in this case where the graph has a small-world structure, apparently adversely affects the performance of MCDCF. Hence, instead of using the Lex ordering as the tie breaking mechanism as we did in our prior work incl. our experiments with random graphs, we have recently started experimenting with selecting a new candidate coalition *uniformly at random* among the equally preferable candidate coalitions. Our early results indicate that random selection of the next candidate coalition tends to result in fewer singleton coalitions and therefore better resulting coalition structure overall. We however leave further discussion until we obtain stronger experimental evidence of this tentative improvement.

5 Summary and future work

We have applied our distributed coalition formation algorithm to network topologies that, to various extents, exhibit the small-world structure. We have tested our algorithm on randomly generated small-world graphs with up to 200 nodes. Our simulations have produced several generally expected results, which to a considerable (but not full) extent validate prior claims that our approach to coalition formation via a particular way of distributed graph partitioning is applicable to a broad variety of underlying graphs [15, 17, 20].

However, we have also obtained some rather unexpected and, in some instances, disappointing results. One lesson learned is that the optimizations of MCDCF reported in [20], while very helpful when the algorithm is run on certain types of graphs such as structure-free random graphs [21], appear inadequate when MCDCF is run on the small-world graphs. Therefore, one immediate objective of our ongoing research is to improve inter-agent coalition negotiation as well as individual agent's candidate coalition search mechanisms, in

order to ensure faster convergence, as well as better resulting coalition structures, when our algorithm is run on small-world-like underlying networks. We also need to investigate in more depth what properties of small-world graphs cause subpar performance of MCDCF with respect to some of the metrics discussed in this paper.

Plans for the future work, in addition to further analyzing and fully understanding some unexpected aspects of the MCDCF performance on the small-world graphs, include studying our algorithm's performance on the *power-law graphs* of comparable sizes and densities to the random graphs we studied in [21] and small-world graphs studied in the present paper. Additionally, we are very interested in experimentally investigating robustness and fault-tolerance properties of our algorithm in the presence of individual node and/or link failures. In that context, it is of a particular interest to understand how the graph structure (for example, Erdos-Renyi vs. small-world vs. power-law graphs) affects robustness of MCDCF. We hope to report new insights on these interesting questions in the near future.

References

[1] S. Abdallah, V. Lesser. *Organization-Based Cooperative Coalition Formation, in* Proc. IEEE / WIC / ACM Int'l Conf. on Intelligent Agent Technology (2004)

[2] N. M. Avouris, L. Gasser (eds.). *Distributed Artificial Intelligence: Theory and Praxis*, Euro Courses Comp. & Info. Sci. vol. 5, Kluwer Academic Publ. (1992)

[3] T. H. Cormen, C. E. Leiserson, R. L. Rivest. *Introduction to Algorithms*, MIT Press (1990)

[4] M. R. Garey, D. S. Johnson. *Computers and Intractability: a Guide to the Theory of NP-completeness*, W.H. Freedman & Co., New York (1979)

[5] M. Jang, S. Reddy, P. Tosic, L. Chen, G. Agha. *An Actor-based Simulation for Studying UAV Coordination*, in 16th European Simulation Symposium (ESS '03), pp. 593-601, Delft, The Netherlands (2003)

[6] X. Li, L.K. Soh. *Investigating reinforcement learning in multiagent coalition formation*, TR WS-04-06, AAAI Workshop Forming and Maintaining Coalitions & Teams in Adaptive MAS (2004)

[7] D. de Oliveira. *Towards Joint Learning in Multiagent Systems Through Opportunistic Coordination*, PhD Thesis, Univ. Federal Do Rio Grande Do Sul, Brazil (2007)

[8] T.W. Sandholm, V.R. Lesser. *Coalitions among computationally bounded agents*, in *Artificial Intelligence* vol. 94, pp. 99-137 (1997)

[9] T. W. Sandholm, K. Larson, M. Andersson, O. Shehory, F. Tohme. *Coalition structure generation with worst case guarantees*, in *AI Journal* vol.111 (1-2), pp. 209-238 (1999)

[10] O. Shehory, S. Kraus. *Task allocation via coalition formation among autonomous agents*, in Proceedings of IJCAI-95, Montréal, Canada, pp. 655–661 (1995)

[11] O. Shehory, S. Kraus. *Methods for task allocation via agent coalition formation,* in *AI Journal* vol. **101** (1998)

[12] O. Shehory, K. Sycara, S. Jha, *Multi-agent coordination through coalition formation*, in *Intelligent Agents IV: Agent Theories, Architectures & Languages*, LNAI #1365, pp.153-164, Springer (1997)

[13] L. K. Soh, X. Li. *An integrated multilevel learning approach to multiagent coalition formation*, in Proc. Int'l Joint Conf. on Artificial Intelligence IJCAI'03 (2003)

[14] R. Sun. *Meta-Learning Processes in Multi-Agent Systems*, in *Intelligent agent technology: research and development*, N. Zhong, J. Liu (eds.), pp. 210-219, World Scientific, Hong Kong (2001)

[15] P. Tosic, G. Agha. *Maximal Clique Based Distributed Group Formation Algorithm for Autonomous Agent Coalitions*, in Proc. Workshop on Coalitions & Teams, AAMAS-2004, New York City, New York (2004)

[16] Tosic, P., Agha, G. *Maximal Clique Based Distributed Coalition Formation for Task Allocation in Large-Scale Multi-Agent Systems*, Post-Proceedings of MMAS-04; LNAI vol. 3446, pp. 104-120, Springer (2005)

[17] P. Tosic. *Distributed Coalition Formation for Collaborative Multi-Agent Systems,* MS thesis, Univ. of Illinois at Urbana-Champaign (UIUC), Illinois, USA (2006)

[18] P. Tosic, G. Agha. *Understanding and Modelling Agent Autonomy in Dynamic Multi-Agent, Multi-Task Environments*, in Proc. EUMAS '03, Oxford, England (2003)

[19] P. Tosic et al., *Modeling a System of UAVs on a Mission,* session on intelligent agent systems & technologies, in Proc. 7th World Multiconference on Systemics, Cybernetics & Informatics SCI'03 (2003)

[20] P. Tosic, N. Ginne. "MCDCF: A Fully Distributed Algorithm for Coalition Formation in Collaborative Multi-Agent Systems", Proc. WorldComp ICAI'10, Las Vegas, Nevada, USA (2010)

[21] P. Tosic, N. Ginne. "A Scalable Distributed Graph Algorithm for Dynamic Coalition Formation: A Performance Case Study on Random Graphs". Technical Report (2011)

[22] G. Weiss (ed.) *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence,* MIT Press (1999)

[23] M. Wooldridge. *An Introduction to Multi-Agent Systems*, Wiley (2002)

[24] The Watts-Strogatz small-world graph generator that we used can be found at http://networkx.lanl.gov/reference/generated/networkx.generators.random_graphs.watts_strogatz_graph.html

[25] Duncan J. Watts. "The "New" Science of Networks", *Annual Review of Sociology*, vol. 30, pp. 243-270 (2004)

Improved Ant Clustering Algorithm

B. Chandra, Abhishek Karpade and Piyush Mehta

***Abstract*- Clustering has applications in variety of fields like customer segmentation, credit card fraud detection and medical imaging. Several clustering techniques exist in the literature. Ant Based clustering [2,4] is intended to improve the quality of clusters as compared to other clustering techniques. However Ant clustering takes lot of time in order to produce good quality clusters. In this paper, Improvements over existing Ant Based clustering algorithm has been proposed by forming a dissimilarity matrix, creating a list of agents based on the sorted dissimilarity values, proper placement of agents and the choice of agents in each iteration based on dissimilarity values in the ant clustering algorithm.**

It has been demonstrated on fifteen benchmark datasets that the proposed method improves the quality of clusters and reduces the computational time as compared to the standard Ant Clustering algorithm and the widely used K means algorithm.

Keywords: Ant clustering, dissimilarity values, agents

I. INTRODUCTION

Clustering [3.5] is a data mining technique in which data is partitioned into groups of similar objects. Widely known clustering algorithms include K-means [6] and Hierarchical clustering [3]. K-means is one of the popular clustering algorithms but is has a drawback that the number of clusters have to be known apriori. In Hierarchical clustering algorithm, one has to fix the threshold in order to determine the number of clusters which is subjective in nature. Ant based clustering algorithm [2] is used for finding good quality clusters. Though ant based clustering algorithm takes lot of time as compared to other clustering algorithms to form clusters, still many analysts prefer this algorithm due to the quality of output. The standard ant clustering takes around fifteen percent of agents in every cycle which make the algorithm slow. Most of the existing results show that the complexity of the algorithm increases with the number of items being clustered. While many other clustering algorithms are superior for small data sets, this ability of ant-based clustering to scale with only the number of items is promising for large data sets.

B. Chandra is Professor at the Department of Mathematics, Indian Institute of Technology, New Delhi, 110016, India (corresponding author phone: 91-11-26591493 ; e-mail: bchandra104@yahoo.co.in).
Abhishek Karpade is an MTech Student in the Department of Industrial and Management Engineering, Indian Institute of Technology, Kanpur, India
Piyush Mehta is Associate Professor at the Department of Industrial and Management Engineering, Indian Institute of Technology, Kanpur, India.

So far, the researchers have contributed only to accelerate the algorithm; there is no contribution in cluster refinement. A novel algorithm has been proposed in this paper to improve the clustering quality of clusters and enhance the speed. The proposed algorithm consists of four steps; in the first step, the initial agents for clustering are selected based on the highest dissimilarity values between items and the number of agents will be taken from the sorted list of dissimilarity values. In the second step, the agents are placed away from each other on a 2D grid. In the third step, remaining items are placed nearby those agents based on their smallest dissimilarity values. In the fourth step, we follow the standard ant clustering algorithm with the change in selected initial agents. We have proposed a novel method to improve the execution time of ant based clustering algorithm which also improves the quality of clusters.

The algorithm has been tested on fifteen datasets taken from UCI machine learning repository [7] and TREC [8]. The results of Improved Ant clustering on all the datasets was compared with the results of standard Ant clustering algorithm and widely used K means clustering. Based on the performance measures like f-mean and Entropy [1,9] it was found that the Improved Ant clustering not only enhances the quality of clusters but also reduces the computational speed.

II. ANT CLUSTERING

Ant Based clustering [2] is modelled based on the concept that ants can recognize the difference between objects of two or more types with some degree of error. Ants gather items (dead insects) to form heaps and cluster by properties. It clusters data items based on the following two rules. When an ant moves towards an object if the object can be isolated more probability of picking up the object will be greater. When an ant moves an object from one place to another place if the object is surrounded by same types then the probability of dropping it down is greater. This algorithm clusters the dataset without demanding message passing among the ants or any global description of the domain.

Firstly data items and agents are randomly arranged over the grid. One data item is picked randomly by each agent. Whether to drop its data item or not is probabilistically decided by the agent. The probability of dropping down is depends upon the number of objects of same type in that local neighborhood. The probability of dropping down an item using neighborhood function f(i) can be calculated as follows:

$$P_{drop}(i) = \left(\frac{f(i)}{k^- + f(i)}\right)^2 \quad (1)$$

where K^- is taken to be 0.3, f is defined as an estimation of the fraction of nearby points occupied by the objects of the same type and it can be computed as

$$f(i) = \max\left(0, \quad \frac{1}{\sigma^2}\sum_{j \in L}\left(1 - \frac{\delta(i,j)}{\propto}\right)\right) \tag{2}$$

Where,

σ^2 is the size of the local neighbourhood L ($\sigma^2 \in \{9, 25\}$).

$\delta(i,j)$ Is a dissimilarity function $(\delta \in [0,1])$.

$\propto$ Is data dependent scaling parameter $(\propto \in [0,1])$.

If a drop decision of a data item occurs then agent drops the data item at the grid location and it attempts to find another data item to pick up.

$$P_{pick}(i) = \left(\frac{k^+}{k^+ + f(i)}\right)^2 \tag{3}$$

Where K+ = 0.1 and f is the same as in equation (2).The agent randomly selects a data item i, on the basis of density of data items in the local neighborhood of i. The search continues until a data point successful picks up and then the loop repeats with another agent.

III. IMPROVED ANT CLUSTERING

A novel algorithm has been proposed to improve the performance of standard ant clustering both in terms of cluster quality and computational time. The standard ant clustering takes long time for forming the clusters. There are some drawbacks with standard ant clustering which are as follows

(a) Placing items randomly on the 2D grid plane, which may create placing of items diversely.

(b) The standard ant clustering takes large number of iterations to form the clusters due to the drawback mentioned in (a)

(c) The standard ant clustering required taking 15% of agents in each iteration. In this case there is a high probability of choosing some of the agents which have already been chosen in the previous iteration.

In the proposed algorithm, we try to overcome these issues by modifying standard ant clustering. Various steps involved in the modified algorithm are given below:

A. Agent Selection in Initial Phase

In the initial phase, the dissimilarity function values between items is computed for all the items and the dissimilarity matrix is formed. Any distance measure can be chosen but for the present study, Euclidean distance is chosen. The elements in the dissimilarity matrix are normalized between 0 and 1. We sort these values in descending order and create a list of items from it. The items which are farthest apart from each other on data space will be selected as agents in the initial phase.

B. Items Placing on 2D Grid Plane

In the second phase the selected agents from the previous phase are placed on the 2D grid plane. The agents are placed far away from each other on the plane. The remaining items from the dataset are placed near the agent that has the smallest dissimilarity with them as compared to other agents. The size of the grid plane is (x,y) = $(\sqrt{10} \times \sqrt{Items}\ , \sqrt{10} \times \sqrt{Items}\ )$**.** Items represents total number of items in the dataset.

C. Finding Pick up and Drop Probability

The input to this phase is the 2D grid plane with items and agents chosen in the previous phase. Now the five percent agents that are initially chosen will get active. Every agent will check the pick probability of that particular item on which it is placed on the 2D grid plane. If the pick probability for that item is more than 0.9 then it will be picked by the agent. For the picked item, the agent will search for free cell near that cell and check the drop probability. If the drop probability for that item is more than 0.9, then the agent will drop that item on the free cell. Otherwise, the agent holds it till the next iteration and again check for drop probability of that item. However, if the pick probability of item is less than 0.9, then the agent will get deactivated in next iteration.

D. Choice of Next set of Agents

In this fourth phase, the next iteration begins but this time next five percent agents from the Agents list is picked up. Percentage of next set of agents from the list is dependent on the number of agents still holding the items. Steps three and four are repeated. The number of iterations depends on the parameter settings.

Step by step execution of improved ant clustering is given as follows:

1. Normalize the attributes values and store the items with the normalized attribute values in a two dimensional array. This array is refers to as Items array.
2. Find the dissimilarity values between items in the Items array. The computed dissimilarity values are stored in a dissimilarity matrix.
3. Sort the dissimilarity values in descending order and store the values with items in an array called Agents array.
4. Create 2D grid of size (x,y) = $(\sqrt{10} \times \sqrt{Items}\ , \sqrt{10} \times \sqrt{Items}\ )$**.** Five percent agents are

selected from the Agents array and are placed far from each other on the 2D grid plane.

5. After placing five percent agents the remaining items will be placed near one of the agents. The items will be placed near the agent that has smallest dissimilarity value with that item.
6. Now the agents on the 2D grid plane will get active. The pick probability (P_{pick}) is computed using equation (1) and the neighborhood function (f(i)) is found using equation (2). If the pick probability of that item is greater than 0.9 then the agent will pick the item and proceed to step 7 else to step 9.
7. After picking the item the agent will look for a free cell where this item can be dropped. At this point in time the agent will check the drop probability (P_{drop}) of the item using equation (1) and the neighborhood function (f(i)) using equation (2). If the drop probability is more than 0.9, then the agent will drop the item on that free cell and proceed to step 8 else go to step 9.
8. Similarly the remaining agents will follow step 6 and 7.
9. In the next iteration next five percent agents will be selected from the Agents list. The steps 6 to 9 are again followed. During the selection of next set of agents the algorithm will also check which agents are still holding items since, these agents will also be considered in next agents selection procedure.

The steps shown in the form of flowchart is given in the Appendix.

IV. COMPARATIVE PERFORMANCE EVALUATION

The performance of Improved Ant clustering algorithm (IAC) was tested on fifteen different datasets taken from different domains. The source of these datasets is UCI machine learning repositories [7] and TREC [8] which consists of benchmark datasets. Clustering on these datasets was also done using K-means clustering and standard Ant Based clustering. The performance measures used are entropy and f-measures [1,9]. Comparative performance in terms of f-measure is given in Table 1 and the same is depicted in Figure1.

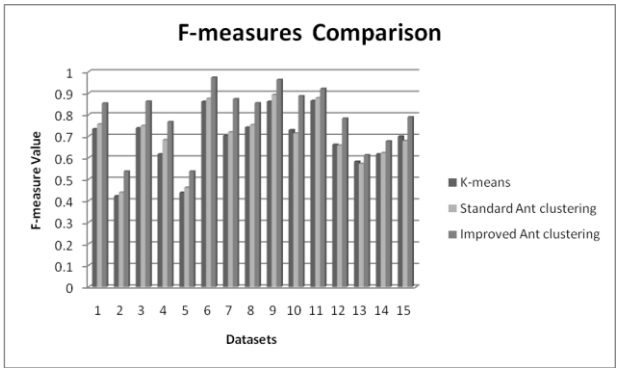


Fig 1.

In the Figure 1, x-axis represents datasets and y-axis represents f-measures values for all the datasets on all the three algorithms. From this Figure and the values in Table 1 we find that the f-measures values of Improved Ant clustering is higher as compared to other clustering algorithms, which indicates that the cluster quality is high in case of Improved Ant clustering.

TABLE I

f-MEASURES VALUES OF CLUSTERING ALGORITHMS

	k-means	AC	IAC
Water plant	0.734	0.756	0.853
Breast cancer	0.422	0.439	0.537
Dermatology	0.738	0.749	0.862
Ecoli	0.617	0.683	0.767
Lung cancer	0.438	0.462	0.537
Fa2	0.861	0.875	0.973
Wine	0.703	0.719	0.873
Sports	0.741	0.753	0.854
Pendigit	0.861	0.893	0.963
Hitech	0.729	0.714	0.887
Xaa	0.865	0.878	0.921
Ra32	0.661	0.658	0.783
Image segmentation	0.582	0.573	0.613
Glass	0.617	0.623	0.677
Blood transfusion	0.699	0.679	0.789

Comparison in terms of entropy is shown in Table 2 and graphically in Figure 2.

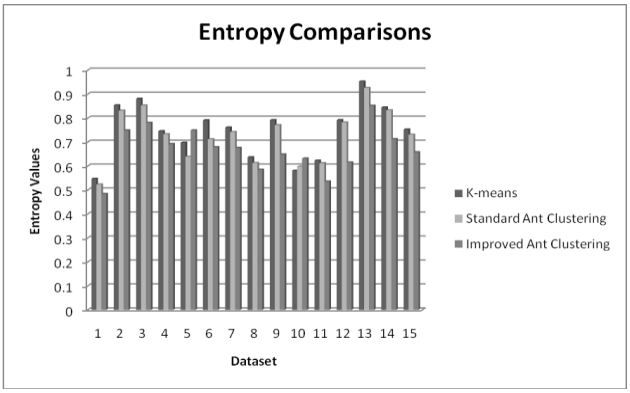


Fig 2

TABLE II

1 ENTROPY VALUES OF CLUSTERING ALGORITHMS

	K-means	**AC**	**IAC**
Water plant	0.547	0.523	0.484
Breast cancer	0.854	0.831	0.749
Dermatology	0.881	0.853	0.781
Ecoli	0.746	0.733	0.692
Lung cancer	0.698	0.639	0.749
Fa2	0.791	0.712	0.679
Wine	0.761	0.742	0.676
Sports	0.637	0.613	0.585
Pendigit	0.792	0.771	0.649
Hitech	0.581	0.598	0.632
Xaa	0.623	0.612	0.536
Ra32	0.792	0.782	0.615
Image segmentation	0.953	0.926	0.852
Glass	0.845	0.833	0.713
Blood transfusion	0.753	0.731	0.658

TABLE III

COMPUTATIONAL TIME TAKEN USING CLUSTERING ALGORITHMS

	K means	**AC**	**IAC**
Water plant	1.021	1.032	0.838
Breast cancer	0.783	0.796	0.632
Dermatology	0.352	0.356	0.235
Ecoli	0.283	0.267	0.125
Lung cancer	0.106	0.102	0.073
Fa2	0.079	0.073	0.051
Wine	0.917	0.193	0.126
Sports	0.782	0.793	0.487
Pendigit	0.028	0.023	0.013
Hitech	1.13	1.21	0.972
Xaa	0.749	0.736	0.643
Ra32	0.085	0.081	0.065
Image segmentation	0.828	0.834	0.748
Glass	0.747	0.739	0.547
Blood transfusion	0.758	0.723	0.419

From Figure 2 and Table 2, we find that the entropy values of Improved Ant clustering are lower as compared to other clustering algorithms and this indicates that the cluster purity is high in case of Improved Ant clustering as compared to other algorithms.

Pairwise t-test also shows that even at one percent level of significance, the f-measure values using the Improved Ant clustering are higher as compared to other clustering algorithm and the entropy values are lower for the Improved Ant clustering.

The values in Table 3 and Figure 3 shows that the Improved Ant clustering not only improves the cluster quality but also reduces the computational time. Pairwise t-test authenticates the fact that the computational time taken by Improved Ant clustering is much lower as compared to other algorithms.

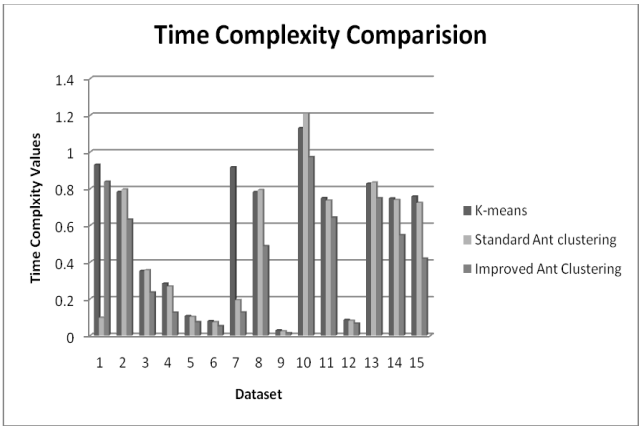


Fig. 3

By comparing the performance of Improved Ant clustering with K-means and standard Ant clustering, it is seen that the proposed algorithm not only improves the cluster quality but also reduces the computational time drastically.

V. CONCLUSIONS

A new modified Ant Based Clustering algorithm has been proposed in this paper .The improvements are in terms of choice of agents, number of agents selected in each iteration and the placement of agents. The agents are selected using Agents list prepared from the sorted values of dissimilarity matrix. The proposed algorithm gives better cluster quality and at the same time reduces the computational time. The efficiency of the Improved Ant clustering has been tested on fifteen benchmark datasets.

REFERENCES

[1] T.Cover and J.Thomas,"Elements of Information Theory", Wiley ,1991.

[2]. J. L Deneubourg,, S.Goss, N.Franks, A. Sendova-Franks, ,C. Detrain, L.Chrétien, "The dynamics of collective sorting: Robot-like ants and ant-like robots", In Proceedings of the First International Conference on Simulation of Adaptive Behaviour, From Animals to Animats 1, p. 356-365, Cambridge, MA: MIT Press, 1991.

[3] J. Han and M. Kamber, "Data Mining: Concepts and Techniques", 2nd edition, Morgan Kaufmann, 2006.

[4] J.Handl, J. Knowles and M. Dorigo "Ant-Based Clustering and Topographic Mapping". IRIDIA, 2004.

[5] A.K.,Jain M.N. Murtyand P.J. Flynn. "Data Clustering: A Review", ACM computing survey, 1999.

[6] J. MacQueen,"Some methods for classification and analysis of multivariate observations," In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. Volume I, Statistics. Edited by Lucien M. Le Cam and Jerzy Neyman. University of California Press, 1967.

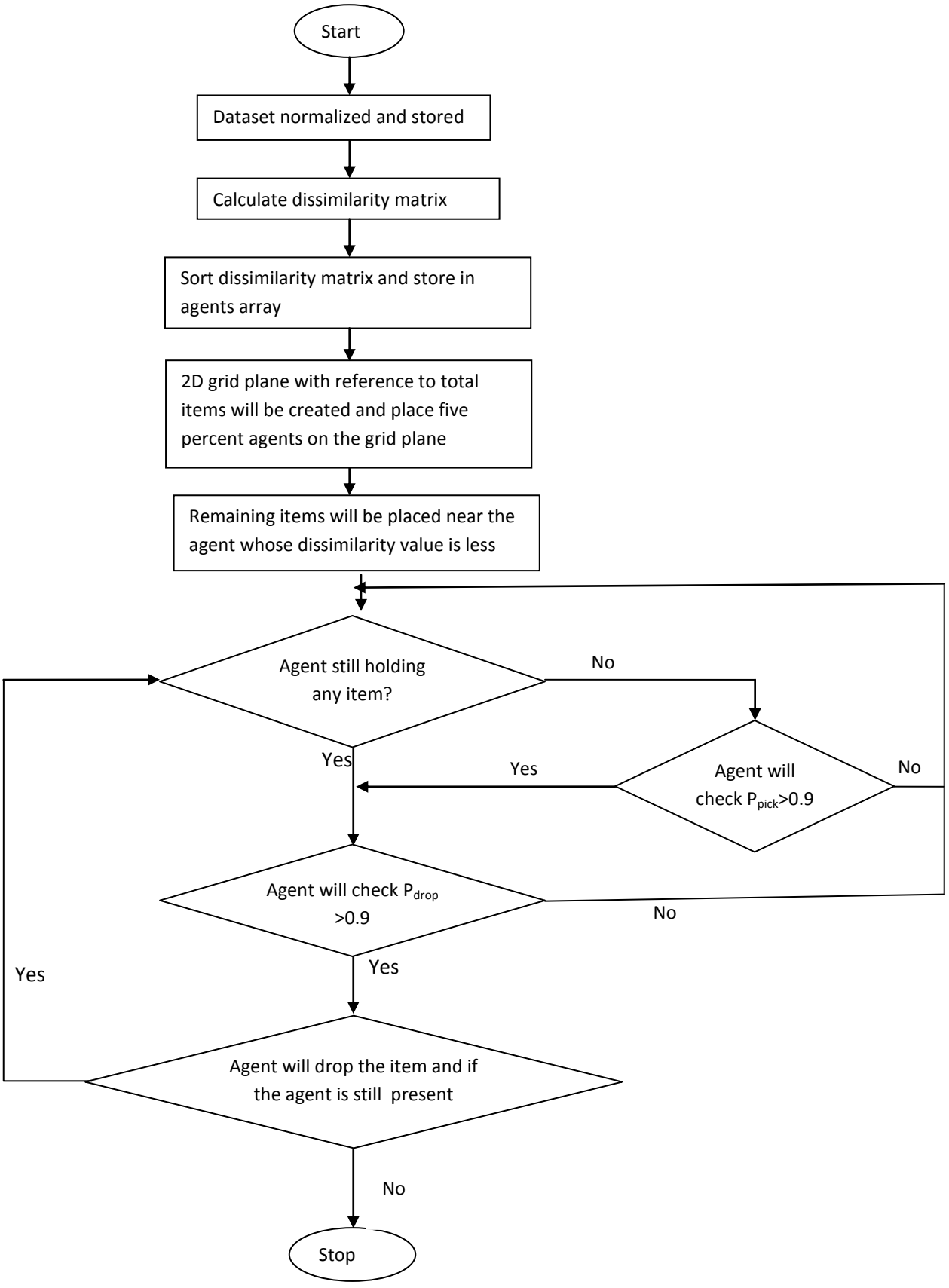
[7] D.Newman, S. Hettich, C. Blake and C.,Merz, UCI Repository of Machine Learning Databases, [Online]. Available: http://www.ics.uci.edu

[8] TREC, Text Retrieval Conference, [Online]. Available: http://trec.nist.gov

[9] H.Xiong, Wu, J.Chen "K-Means Clustering Versus Validation Measures: A Data-Distribution Perspective". IEEE,2009.

APPENDIX

FLOW CHART OF ALGORITHM



Emotional Agents Outreach: An Undergraduate Research project

Bethel Tarekegne, Hong Jiang, Cedrik Brown and Kory Griggs
MSC 2280, School of STEM, Benedict College, Columbia, SC, USA
bettiye@gmail.com

Abstract— *Emotions are psychological reactions that develop as a consequence of certain cognitions and interpretations. In this research we will introduce the OCC (Ortony, Clore, and Collins) theory as our model for understanding emotions. The use of advanced emotional agents has been used on younger kids in a form of video games and animated movies. But the use of this techonology in the teaching learning environment has not yet been popular. So, in our research we will introduce the use of agents in teaching learning environments in a way they would attract and inspire younger kids to be more interested in studying. We will be using Scratch, a programming language used to create agents in a simpler way, as our tool. After this research, with further modification and studies, we will be able to get an updated educational system which involves emotional agents that will help students attain their intended success.*

Keywords: Emotional Agents, Emotion Theory, Decision Making, Emotional Believability

1. Introduction

Emotions are not just things felt inside and expressed on one's face, they go beyond that. They have the power to affect that individual's decision making, and the response of others around. For instance, let us take kids; they react fast for emotional facial reactions better than spoken words. Just by looking at the facial expression on their teachers or parents they try to fix whatever fault they have done to get rid of the consequence that might come next. So we see how one's emotion can affect the response of the other person around. Not only that, but for instance, if someone goes to a shop planning to buy a PS3, and finds out, at the store, that xbox360 is much better, then the person will end up buying the thing he have not planned to buy. From this we understand how it affects one's decision making. So emotions play a great role in everybody's day to day life.

This is the reality of emotions on humans, but the main issue now is how to implement this fact of emotions on agents. The advancement of Artificial Intelligence (AI) has given us advancements in creating emotional agents. And these emotional agents have been improving their believability, but still they have that robotic feature that restricts their decision making being influenced by their emotion. So, finding a solution to minimize this unbelievable characteristic and create a more advanced emotional agent is what is interesting nowadays.

Nevertheless, our research concentrates on applying the emotional agent and introducing it to younger kids like K-12 kids. Nowadays students in the elementary and high school level are familiar with the concept of emotional agents. Video games, computer games, and animated movies these all have done their part to make the kids familiar to this phenomenon called agents. Moreover, thanks to programming languages like Scratch these kids are getting more aware of this issue and in the near future the hope for believable emotional agents and the application of this knowledge in younger kids will become a reality.

Our research will give a good stepping stone for further studies on how to use agents as tutors, study pals, and teaching aids for the kids. We will concentrate more on the introduction part of the idea and we will give ways for future studies. Because it still is an area to be worked on.

2. Research Background

The existence of agents can be traced back to the beginning of artificial intelligence (AI). The AI is a branch of computer science dealing with the simulation of intelligent behaviors in computers. With further studies and researches on AI the creation of agents [1] was performed, and through time these agents were improved and modified to have emotions, and they were called Emotional Agents. The advancement of these emotional agents is also improving the need and study of AI , from which this knowledge first arrived. Any kinds of video games, or computer games, robots, and all actuators, are composed of agents. And this agents need to be emotional in order for the agent to be more realistic. If it is a video game or a computer game then the interaction of the agent is most probably with humans; so, it needs to develop an emotional behavior that could help it interact properly with the human [2], [3], [4].

The first key method for the advancement of emotional agents was having the knowledge to understand emotions very well, and for this there were lots of ideas forwarded from different intellectuals. But for our research we will be introducing the OCC (Ortony, Clore, and Collins) model. The OCC theory gives the right understanding of the detailed emotions available. After the word "emotion" has been understood then it was implemented on the agents and as we said above the knowledge of emotional agents appeared.

Consequently, the next issue was on how to make these emotional agents believable. Humans have emotion and this emotion plays a great deal in their decision making [5], [6], but the effect of emotion on the decision making ability of the agents is not quite similar. This fact shows us that we need further study and work in order to make them believable.

2.1 The theory of Ortony, Clore, and Collins (OCC)

Ortony, Clore, and Collins developed their theoretical approach with the aim to implement it in a computer. They wanted to lay the foundation for a computationally tractable model of emotion. In other words, they wanted an emotion that could in principle be used in an Artificial Intelligence (AI) system that would, for example, be able to reason about emotion. Their theory assumes that emotions develop as a consequence of certain cognitions and interpretations. And they give three aspects that determine these three cognitions which are events, agents and objects [7]. Below Figure 1 is a clarification of what they gave as the link between these three determinants of emotion, and we also assume this as the right understanding of emotion.

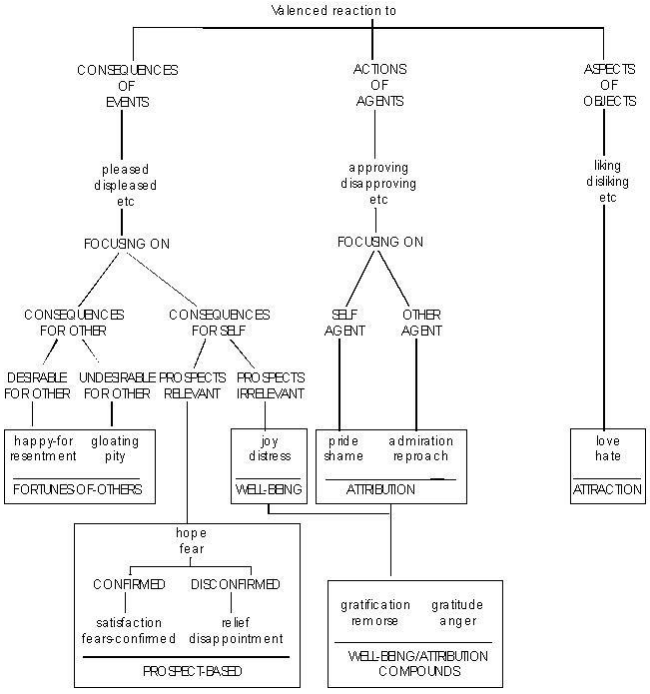


Fig. 1: OCC Model - Determinations of Emotions

2.2 On Making Believable Emotional Agents Believable, Andrew Ortony

The first feature for an agent's emotion to be believable is its consistency and coherence. Consistency relates to the feature of the agents reaction to a certain stimuli to remain constant in similar occasions. For instance, if one person shows a frightened reaction when facing a certain animal today then we expect him to be frightened to that same animal tomorrow. This phenomenon is really important. Moreover, we have coherence together with consistency. The reaction or the emotion one shows for a certain situation should not only apply specifically to that local situation, but it should be applicable in a more global manner [8].

So as to the believability of these emotional agents is concerned, many have done their part to make it more consistent and coherent. But still it is a hot issue because we need to know how to make an agent capable of managing all possible stimuli that it might face, and how to give the right reaction fast enough. So as to the believability of these emotional agents is concerned, many have done their part to make it more consistent and coherent. But still it is a hot issue because we need to know how to make an agent capable of managing all possible stimuli that it might face, and how to give the right reaction fast enough.

3. The Use of Emotional Agents as Tutors on Updating the Educational System

This research mostly concentrates on the formation of believable emotional characters and their application on younger kids. From previous researches we have seen what it means by emotion and their believability. So our work is combining these two with relation to kids and bringing a more useful outcome.

In this twenty-first century the one thing that is in the mind of every younger kid throughout the world is the adaptation of video games. Not long ago if a teacher asks his or her students for their hero they might give names like Albert Einstein, Thomas Edison, Martin Luther King, Jr., and many more in their areas of studies. Nevertheless kids of this generation might not even know who this people are. They are not introduced to any of them; as a result, they are not their heroes. It is either a video game character or an animated movie character that they might give you as their hero.

The above argument brings us to the main reason why we chose to do our research on this topic. We need these younger kids; we need them to be the future scientists, politicians, artists, economists, doctors, engineers, and so on. You can name it whatever you want, but the important thing is that we need to build this generation in order to create a better world in the future. With this in mind let's introduce emotional agents as teaching tools to these younger kids.

Many younger boys and girls, generally 90% of the kids between the age of 11-17, spend much of their time playing basketball video games on xbox360 or PS3, but if we have the SpongeBob the cartoon character or Kobe Bryant the basket ball legend video game character as their tutor then

these kids will for sure spend much of their time studying math, history, or any other subject.

Education by itself is related to our emotions. We have to be emotionally inclined to the subject in order to like it, and if we do not like it then our performance goes down. So what we are trying to say is the method of the teaching system needs to be updated. Emotional agents can be a great use in this regard.

Scratch [9] is one of the programming languages that exist today which can be used in this situation. This program does not need much knowledge on any programming languages. It has its own tutorial lessons that could help the user learn how to use them; so, either parents or teachers can create tutorial lessons with the use of the emotional agents featuring the kid's favorite characters as supplementary teaching materials [10]. Not only that but even the kids can use Scratch to create their own their study pals using their favorite characters or even using their own imagination. This way the students will be excited and more eager to learn their lessons. For instance in the following images we will see two sports legends that happen to be idols for basketball and soccer fans respectively.

Fig. 2: Sample Programs with Sports Legends for Tutoring

The above two images in Figure 2 were copied from a program that we did using Scratch. It is a pre tutorial introduction and the program keeps on with the tutorial lessons and a break time for the kids to play games in which they can practically grasp the knowledge that they gained theoretically.

Creating the awareness and introduction of this interesting development is being done by some summer science camps in different areas. Even though it might take a while before this knowledge reaches everywhere, it is growing in a promising speed. By creating the awareness starting from preschool teachers and parents we can decrease the time it takes to get it advanced. The main aim should not only be introducing it as a way to entertain the students but rather in a way to use it as a teaching tool; with this we can get a potentially fit educated generation.

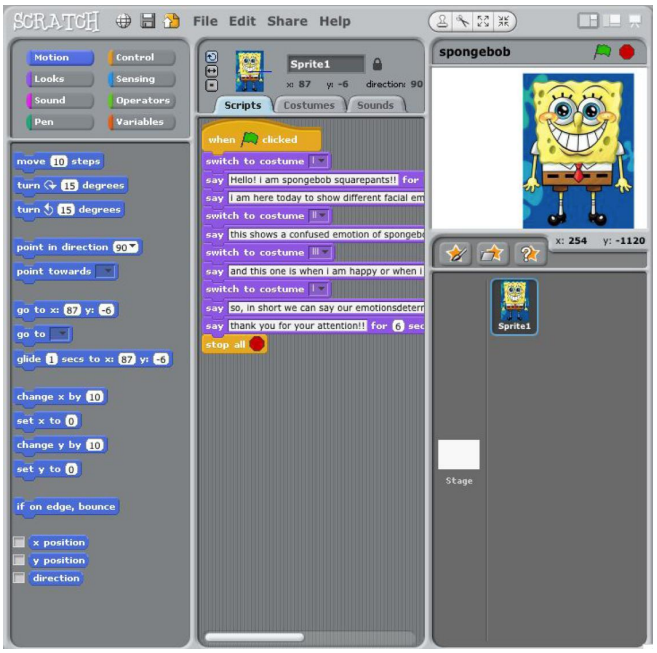


Fig. 3: SpongeBob Tutorial Program

The above Figure 3 illustrates how the Scratch program can be used to form the tutors. In this case SpongeBob, one of the kids' favorite characters, is used as an agent to tutor on a certain topic. The resulted program includes the SpongeBob with different emotional facial expressions, and some are listed in the following Figure 4.

After the agent is given its task it will be ready to give the tutorial for the student; nevertheless now the question becomes what will happen if the student asks a question? This is where we need the believability of the emotional agent. We need our agent's decision making to answer the question to be correct and believable. Since it is a teaching learning environment the agent is supposed to have coordination with the student. To answer this question we were forced to use other algorithms than the once present on Scratch. But since this is a short research and the result is going to be applied on people who do not have the knowledge of forming algorithms it is better to keep it simple and just introduce Scratch and its applications.

Fig. 4: Resulted Tutorial Program with Different Facial Expressions

In general, forming an agent whose decision making is dependent on its emotions and is able to make the human computer interaction (HCI) lively and realistic is a whole accomplishment and fulfillment.

4. Conclusions

This paper has attempted to address the possibility of incorporating emotional agents in the educational environment to enlighten younger kids in a more inspiring and attracting manner. It started by defining emotions in general to making them believable and useful on agents. Then it stated how to introduce the usage of emotional agents on the teaching and learning environment with the use of Scratch as a programming language.

Future studies should be done in order to make it more applicable. With future work the agents' decision making will be improved and they will become more believable. Not only that but future works are needed to form algorithms which are easy to introduce for the society; so that they can create a more developed agents in programming languages like Scratch [11].

References

[1] M. Wooldridge and N. R. Jennings, "Intelligent agents: Theory and practice," *Knowledge Engineering Review*, vol. 10, pp. 115–152, 1995. [Online]. Available: http://www.csc.liv.ac.uk/ mjw/pubs/ker95.pdf

[2] A. Camurri and A. Coglio, "An architecture for emotional agents," *IEEE Multimedia*, vol. 5, pp. 24–33, 1998.

[3] B. Becker, "Social robots - emotional agents: Some remarks on naturalizing man-machine interaction," *International Review of Information Ethics*, vol. 6, pp. 37–45, 2006.

[4] K. A. Maria and R. A. Zitar, "Emotional agents: A modeling and an application," *Inf. Softw. Technol.*, vol. 49, pp. 695–716, July 2007. [Online]. Available: http://dx.doi.org/10.1016/j.infsof.2006.08.002

[5] J. jules Ch. Meyer, "Reasoning about emotional agents," *International Journal of Intelligent Systems*, vol. 21, pp. 601–619, 2006.

[6] H. Jiang and J. M. Vidal, "From rational to emotional agents," in *Proceedings of the AAAI Workshop on Cognitive Modeling and Agent-based Social Simulation*, 2006. [Online]. Available: http://jmvidal.cse.sc.edu/papers/jiang06b.pdf

[7] B. R. Steunebrink, M. Dastani, and J.-J. C. Meyer, "The occ model revisited," in *Proceedings of the 4th Workshop on Emotion and Computing - Current Research and Future Impact*, Paderborn, Germany., 2009.

[8] A. Ortony, "On making believable emotional agents believable," *Emotions in Humans and Artifacts*, pp. 189–212, 2003.

[9] M. Resnick, J. Maloney, N. Rusk, E. Eastmond, A. Millner, J. Silver, E. Rosenbaum, K. Brennan, and A. Blanton, "Scratch," Lifelong Kindergarten group, MIT Media Lab. http://info.scratch.mit.edu/.

[10] B. florin Marin, A. Hunger, and S. Werner, "Corroborating emotion theory with role theory and agent technology: a framework for designing emotional agents as tutoring entities," *Journal of Networks*, vol. 1, pp. 29–40, 2006.

[11] K. Shihab, "Emotional agents in computer games," *International Journal Of Computers*, vol. 3, 2009. [Online]. Available: http://www.naun.org/journals/computers/ijcomputers-139.pdf

Developing a Concept Extraction System for Turkish

Meryem Uzun-Per[1], **Hidayet Takçı** [2], **and Tunga Güngör**[1]
[1] Computer Engineering Department, Boğaziçi University, Bebek, Istanbul, Turkey
[2]Computer Engineering Department, GYTE, Gebze, Kocaeli, Turkey

Abstract - *In recent years, due to the vast amount of available electronic media and data, the necessity of analyzing electronic documents automatically was increased. In order to assess if a document contains valuable information or not, concepts, key phrases or main idea of the document have to be known. There are some studies on extracting key phrases or main ideas of documents for Turkish. However, to the best of our knowledge, there is no concept extraction system for Turkish although such systems exist for well-known languages. In this paper, a concept extraction system is proposed for Turkish. By applying some statistical and Natural Language Processing methods, documents are identified by concepts. As a result, the system generates concepts with 51% success, but it generates more concepts than it should be. Since concepts are abstract entities, in other words they do not have to be written in the texts as they appear, assigning concepts is a very difficult issue. Moreover, if we take into account the complexity of the Turkish language this result can be seen as quite satisfactory.*

Keywords: Concept Extraction, Natural Language Processing

1 Introduction

There is a rapidly growing amount of available electronic information such as online newspapers, journals, conference proceedings, Web sites, e-mails, etc. Using all these electronic information, controlling, indexing or searching is not feasible and possible for a human. For search engines, users have to know the keywords of the subject that they search, since search engines use a top down approach in order to find information in textual materials. The necessity of analyzing unstructured texts automatically is apparent. Users do not have to know the query terms and the main idea of the searched documents. If the concept of a document is known, a general knowledge about it also is known.

Concept is a term related to philosophy more than linguistics. In philosophy, a concept is defined as a thing apprehended by human thought and concepts are elements of thoughts and facts [1]. Concepts are different from words. Words are used for naming the concepts. It is possible that a single word can correspond to more than one concept or several words can define a single concept.

Concept extraction study aims at obtaining efficient solutions to some problems that are harder to solve using data mining. Crangle et al. define concept extraction as follows [2]:

> "Concept extraction is the process of deriving terms from natural-language text that are considered representative of what the text is about. The terms are natural-language words and phrases which may or may not themselves appear in the original text."

For concept extraction from unstructured texts there are mainly two approaches: expert-based and statistical. Expert-based approach has several disadvantages such as finding specialists on the subjects and developing learning based systems. In statistical approach, statistical methods are applied to the training data and models are built. Bayesian networks, neural networks, support vector machines, and latent semantic analysis are some of the statistical methods used in this area. Natural Language Processing (NLP) is different than these approaches in the sense that it uses the speed and cost effectiveness of the statistical approach but sometimes may require human intervention [3]. For linguistics-based approaches human intervention may be needed at the beginning to develop dictionaries for a particular industry or field of study. However, it has several considerable advantages such as getting more precise results quickly. Concepts can be extracted by using these models.

For English there are some studies done for concept extraction such as the studies of Crangle et al. [2] and Gelfand et al. [4], and there are some commercial software such as PASW Text Analytics and WordStat. These software also support several other languages such as French, Italian and Spanish. Moreover, there are some studies for unstructured Turkish documents for key phrase extraction such as [5] and [6]. However, key phrase extraction is different from concept extraction in the sense that key phrases are written in the documents as they appear, but concepts do not have to appear in the documents. There is neither a study on concept extraction nor software for Turkish. In this paper, a concept extraction system for Turkish is proposed.

2 Related Work

Concepts can be formed of words or phrases. Initially, sentences are divided into their words and phrases. For this

purpose, grammatical and syntactic methods are used which are tested in ontology learning, lexical extraction, and information retrieval systems [7]. In grammatical methods, if shallow parsing is used to parse the sentences, the whole sentence is converted into a grammatical tree where the leaves are noun and verb phrases. Then, noun phrases are selected as concepts. In syntactic methods punctuation and conjunctions are used as divisors. Then, all phrases are regarded as concepts. This approach is also used in keyword extraction systems [8].

For concept extraction there are two important application areas which are indexing documents and categorizing documents. Moreover, it is used for evaluating open ended survey questions [9], mapping student portfolios [7], extracting synonymy from biomedical data [2], extracting legal cases of juridical events [10], and several other areas. The main reason of the use of concept extraction in numerous fields is that concepts give an opportunity to enhance information retrieval systems [11, 12].

Extracting key phrases of documents is related to extracting concepts of documents. In academic articles, generally, key phrases are listed after the abstract which help the reader to understand the context of the documents before reading the whole document. Keyphrase Extraction Algorithm (KEA) is an automatic keyhrase extraction method that is proposed by Witten et al.[8]. The KEA was applied to Turkish documents by Pala and Cicekli by changing the stemmer and stop-words modules, and by adding a new feature to the algorithm [5]. Both for English and Turkish the success rates are about 25-30%.

In automatic key phrase extraction field a study is performed by Wang et al. [13] which uses neural networks for extracting key phrases. Turney uses two algorithms to extract key phrases from documents [14]. One of them is the C4.5 algorithm and the other is the GenEx algorithm. The overall success rate is very low. Rohini presented a study that extracts key phrases from electronic books by using language modeling approaches [15]. Kalaycılar and Cicekli [6] proposed an algorithm called TurKeyX for Turkish in order to extract key phrases of Turkish documents automatically which is based on statistical evaluation of noun phrases in a document. A study about extracting concepts automatically from plain texts is done by Gelfand et al. [4] by creating a directed graph called semantic relationship graph from WordNet. The success rate of all these studies is at most 30%.

There is some commercial software which is related to concept extraction. The two most popular software are PASW Text Analytics [3] and WordStat [16]. In Text Analytics linguistic resources are arranged in a hierarchy. At the highest level there are libraries, compiled resources and some advanced resources. Moreover, for English, there are specialized templates for some specific application areas like gene ontology, market intelligence, genomics, IT, and security intelligence. There are two types of dictionaries in libraries: compiled dictionaries which end users cannot modify and other dictionaries (type, exclusion, synonym, keyword, and global dictionaries) which end users can modify. The compiled dictionaries consist of lists of base forms with part-of-speech (POS) and lists of proper names like organizations, people, locations and product names. After extracting candidate terms, named entities and the dictionaries are used to identify concepts of documents. WordStat also uses the same principal while extracting concepts of the texts.

3 Concept Extraction System

3.1 Pre-processing

In order to develop a Concept Extraction System (CES) for Turkish, a corpus has to be determined to work on. The first step in this work is finding comprehensive Turkish documents. Then the pre-processing processes start. In order to run the codes on documents, all the documents have to be converted to text format. The text files are saved in UTF-8 format. Then, all documents in the corpus are tokenized such that a blank character is inserted before the punctuation characters.

3.2 Creating nouns list

Concepts can be determined from the nouns and the noun phrases. Therefore, in order to obtain the concepts of the documents, nouns in the documents have to be extracted. Extracting nouns of the documents and eliminating inflectional morphemes are difficult issues for Turkish. In this process, The Boun Morphological Parser (BoMorP) and The Boun Morphological Disambiguator (BoDis) programs [17] are used. They parse documents with an accuracy of 97%. They are applied to all the documents in the corpus. BoMorP parses the words and identifies their roots and morphemes. Turkish words are highly ambiguous in the sense that a single Turkish word can have several distinct parses. BoDis calculates a score for each parse according to the context. The output shows the root, the POS tag in square brackets, inflectional morphemes with '+' sign, derivational morphemes with '–' sign, and the score. The parse of an example word is as follows:

tekniklerin (*of the techniques*)
teknik[Noun]+lAr[A3pl]+[Pnon]+NHn[Gen] : 21.4033203125
teknik[Adj]-[Noun]+lAr[A3pl]+Hn[P2sg]+[Nom] : 19.7978515625
teknik[Adj]-[Noun]+lAr[A3pl]+[Pnon]+NHn[Gen]: 14.263671875
teknik[Noun]+lAr[A3pl]+Hn[P2sg]+[Nom] : 12.658203125

After the disambiguation process, the nouns in the documents are selected. If the parse with the highest probability is noun, it is selected unless it is an acronym,

abbreviation, or proper name. These types are also represented as noun in the root square bracket, but in the next square bracket their original type is written. So, the second square bracket is also checked in order to obtain the correct nouns list.

Inflectional morphemes are removed from the nouns. For example, the root forms of all the words "sistem, sistemler, sistemlerin, sistemde, sistemin, sisteme, etc." (*system, systems, of the systems, in the system, of the system, to the system,* etc.) are regarded as "sistem" and their frequencies are added to the "sistem" noun. However, derivational morphemes are kept as they appear. For example, the noun "delik" (*hole*) is derived from the verb "delmek" (*to drill*), however the noun "delik" is added to the nouns list in this form. All nouns are listed for the documents and their frequencies are calculated. Then all nouns are stored in one file, the same words in the documents are merged, and their frequencies are added. The nouns that occur in documents rarely are considered as they cannot give the main idea of them. If the frequencies of the nouns are less than three, they are eliminated to decrease the size of the list and speed up later processing.

3.3 Clustering cumulative nouns list

Concepts can be defined by nouns. Therefore, clustering similar nouns is helpful in order to determine the concepts. For this purpose, some clustering methods such as hierarchical clustering and k-means clustering are applied to the cumulative nouns list. These clustering methods are unsupervised learning algorithms which do not need any training step to pre-define the categories and label the documents.

First of all, document-noun matrix is created from the cumulative nouns list, which holds the documents in rows and the nouns in columns, and the intersection of a row and a column gives the number of times that noun appears in the document. The clustering algorithms are applied to the matrix for different numbers of clusters such as 10, 25, 50, 75 and 100. Hierarchical clustering algorithms are coded by MATLAB, k-means algorithm is applied by Tanagra [18]. Clusters are assessed by human specialists. It is observed that the k-means algorithm for 100 clusters performs much better than the other possibilities.

3.4 Assigning clusters to documents

After the clustering operation, the clusters are assigned to the documents. This is done by searching the nouns of the documents in the words of the clusters. A ratio is calculated for each possible cluster of a document by dividing the number of the words in the possible cluster of the document to the number of the words in that cluster. If the ratio is more than a threshold value, the cluster is assigned to the document. So, it can be said that this document can be defined by that cluster. The threshold is selected as "1"; in other words, if a document contains all the words of a cluster, this cluster is assigned to that document. Because, it is observed that if a document is related to a cluster it should contain all the words of that cluster. More than one cluster can be assigned to a document. Similarly, a cluster can be assigned to more than one document. Figure 1 shows the pseudo-code of assigning clusters to documents.

```
Input
    F1: Documents-Words file
    F2: Clusters-Words file
Output
    F3: Documents-Clusters file
Begin
1:  L1 <- Read F1 to list
2:  L2 <- Read F2 to list
3:  for each word w in L1
4:      Search cluster cl of w in L2
5:      Append cl to L1
6:  end for
7:  for each document d
8:      L3 <- Read clusters of d in L1
9:      L4 <- Read words of d in L1
10:     for each cluster cl in L3
11:         A <- Number of words in cl in L4
12:         B <- Number of words in cl
13:         if (A/B >= Threshold)
14:             Write d + cl to F3
15:         end if
16:     end for
17: end for
End
```

Figure 1: Assigning clusters to documents

3.5 Identifying documents by concepts

The main aim of this study is to define documents with concepts. Therefore, a transition has to be done from words and clusters to concepts. In concept extraction software like PASW Text Analytics and WordStat, dictionaries are used in order to identify documents by concepts as mentioned before [3, 16]. These dictionaries consist of concepts and words related to these concepts. In both programs, users can add or remove concept categories or words to the categories. Similar to these programs, it is decided to create concept categories and words related to them. So, concepts have to be assigned to clusters according to the words they contain by human specialists. Then, concepts are assigned to the documents according to their assigned clusters. Figure 2 shows the module for assigning concepts to documents.

```
Input
    F1: Documents-Clusters file
    F2: Clusters-Concepts file
Output
    F3: Documents-Concepts-Count file
Begin
1:  L1 <- Read F1 to list
2:  L2 <- Read F2 to list
3:  for each document i
4:      L3 <- Read clusters of i
5:      L4 <- empty
6:      for each cluster cl in L3
7:          L5 <- read concepts of cl
8:          for each concept c in L5
9:              if (L4 does not contain c)
10:                 Add c + "1" to L4
11:             else
12:                 Increase count of c in L4
13:             end if
14:         end for
15:     end for
16:     Write L4 to F3
17: end for
End
```

Figure 2: Assigning concepts to documents

4 Experiments and Evaluations

4.1 Corpus Selection

In order to develop a CES for Turkish, a corpus is needed to work on. The first step in this work is finding comprehensive Turkish documents. Online archives of the Journal of the Faculty of Engineering and Architecture of Gazi University [19] are selected as a corpus which is also used in [5] and [6]. It contains 60 Turkish articles and 60 key files which contain the keywords of the articles.

4.2 Testing Methodology

After selecting a corpus the methodology is applied to the corpus by following the steps explained before. Then several tests are applied to the results obtained. These tests are test by words, test by clusters, and test by concepts. Precision and recall are used in order to measure the success rates which are widely used metrics to evaluate the correctness of results of data mining projects. Equations 1 and 2 show the formula of precision and recall, respectively, where *a* is the number of retrieved and relevant records, *b* is the number of retrieved records, and *c* is the number of relevant records [20].

$$precision = \frac{a}{a+b} \quad (1)$$

$$recall = \frac{a}{a+c} \quad (2)$$

4.3 Test by words

Correctness of the clusters which are assigned to the articles is tested by words via the words of the key files. If the clusters are created and assigned correctly, the words in the clusters which are assigned to the articles should match with the nouns in the key files. We denote the words of the clusters which are assigned to an article as *w1* and the nouns in the key file of that article as *w2*. *w2* is searched in *w1*. For each article, the numbers *w1*, *w2*, and the intersection of *w1* and *w2* are calculated. Here precision is not needed to be calculated because clusters contain a lot of words and limiting them is not possible in this methodology. Only recall is calculated. According to Equation 2; *a* is the number of the intersection of *w1* and *w2*, *(a + c)* is the number of *w2*.

Average recall is calculated as 0.46. About half of the nouns of the key files are contained in the words of the clusters which are assigned to the articles. This information cannot explain the accuracy of the study because the clusters contain a lot of words in them; however the words of the key files are very limited. But unfortunately, although a lot of nouns are selected from the articles, only half of them are matched with the nouns of the key files. Figure 3 shows the number of the nouns of the key files versus the number of the matched nouns for the documents.

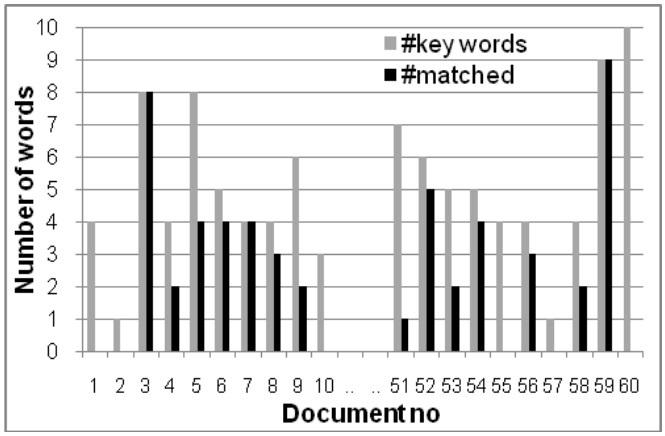


Figure 3: Number of key words versus number of matched words

4.4 Test by Clusters

Correctness of the clusters which are assigned to the articles is tested by clusters via the clusters of the key files. Clusters are assigned to the key files according to the nouns in them. We denote the clusters of an article as *cl1* and the clusters of the key file related to that article as *cl2*. *cl1* and *cl2* are compared. For each article, the numbers *cl1*, *cl2*, and the intersection of *cl1* and *cl2* are calculated. Then, precision and recall are calculated for each document. According to Equations 1 and 2; *a* is the number of the intersection of *cl1* and *cl2*, *(a + b)* is the number of *cl1*, and *(a + c)* is the number of *cl2*.

Average precision and average recall are calculated as 0.50 and 0.41, respectively. As a result of the test by clusters, 41% of the assigned clusters are matched with the clusters of the key files. Half of the clusters which are assigned to the articles are assigned correctly. The recall is lower than expected. Since the clusters are considered as general topics of the articles, it indicates that the general topics of the articles cannot be determined perfectly. However, for Turkish it can be regarded as a success because of the complexity of the language. Figure 4 shows the number of the clusters of the key files versus the number of the matched clusters for the articles.

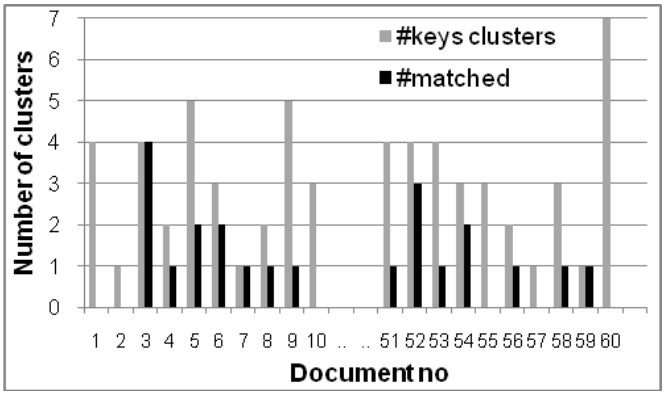


Figure 4: Number of clusters of the key files versus number of the matched clusters

4.5 Test by Concepts

Correctness of the concepts which are assigned to the articles is tested by concepts via the concepts of the key files. We denote the concepts which are assigned to an article as *c1* and the concepts of the key file related to that article as *c2*. *c1* and *c2* are compared. For each article, the numbers *c1*, *c2*, and the intersection of *c1* and *c2* are calculated. Then, precision and recall are calculated for each article. According to Equations 1 and 2; *a* is the number of the intersection of *c1* and *c2*, *(a + b)* is the number of *c1*, and *(a + c)* is the number of *c2*.

Average precision and average recall are calculated as 0.22 and 0.51, respectively. As a result of the test by concepts, 51% of the concepts which are assigned to the articles are matched with the concepts of the key files. 22% of the concepts which are assigned to the articles are assigned correctly. This shows that more concepts are assigned than it should be. The recall being too high may be due to this fact. Since concepts are abstract entities, in other words they do not have to be written in the texts as they occur, assigning concepts is a very difficult issue. Furthermore, Turkish is an agglutinative and complex language that studies on Turkish do not give high scores. For example, the success rate of key phrase extraction studies [5] and [6] are not passed over 30%. As the first study for Turkish in this subject, the results can be seen as quite successful. Figure 5 shows the number of the concepts of the key files versus the number of the matched concepts for the articles.

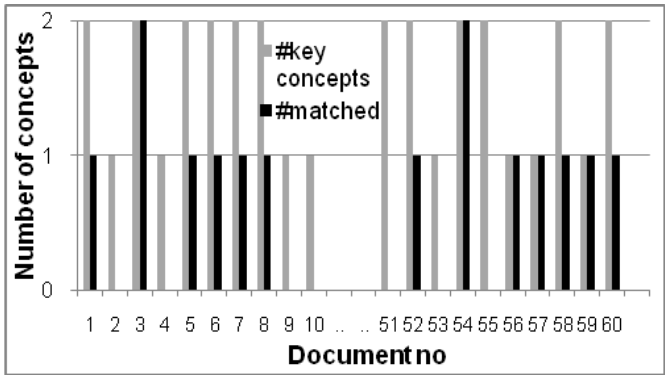


Figure 5: Number of key concepts versus number of the matched concepts

As a result of the test by concepts, precision is considered as low; therefore it is thought that limiting the number of the concepts assigned to the articles may be useful for the results. Due to the similarity of the clusters, some clusters contain the same concepts. So, while assigning concepts to the articles via clusters, some concepts are assigned to the articles more than once. Therefore, we performed another experiment in which a restriction is applied to the concepts of the articles such that if an article is defined by a concept more than once, the concepts that exist only once are eliminated. If an article is defined by concepts only once, no elimination is applied. For evaluation, the same formulas are applied which are explained in the test by concepts. Average precision and average recall are calculated as 0.16 and 0.27, respectively. Both precision and recall decrease significantly. By applying this test, precision is expected to be increased however it decreases. Moreover, recall decreases drastically. If we eliminate all the concepts of the articles which exist only once to define the articles, the results get worse. This shows that the results are much better without any elimination. Therefore, the result of this test can be given as 51% recall with 22% precision.

5 Conclusions

In this paper, a concept extraction system for Turkish is proposed. The first issue that must be faced is the complexity of Turkish which is an agglutinative language. The second issue is the abstractness of concepts. To the best of our knowledge, this study is the first concept extraction study for Turkish. This work can serve as a pioneering work in concept extraction field for agglutinative languages. The results are better than the studies related to this field.

As a future work, the methodology must be applied to new corpora in different domains. In order to improve the methodology, other clustering methods such as supervised learning algorithms can be tried.

6 Acknowledgements

This work is supported by the Boğaziçi University Research Fund under the grant number 5187, the Scientific and Technological Research Council of Turkey (TÜBİTAK) under the grant number 110E162. Meryem Uzun-Per is supported by TÜBİTAK BİDEB 2210. The authors would like to thank to İlyas Çiçekli for the data set.

7 References

[1] Mengüşoğlu, T., *Felsefeye Giriş*, 7. Ed., Istanbul: Remzi Kitabevi, 1992.

[2] Crangle, C., Zbyslaw, A., Cherry, M., and Hong, E. L., "Concept Extraction and Synonymy Management for Biomedical Information Retrieval", in *Proceedings of the 13th Text REtrievel Conference*, Gaithersburg, MD: National Institute of and Technology, 2004.

[3] SPSS Inc. "Mastering new challenges in text analytics", Technical Report, MCTWP-0109, 2009.

[4] Gelfand, B., Wulfekuhler, M., and Punch W.F. III., "Automated Concept Extraction from Plain Text", Technical Report, WS-98-05, AAAI, 1998.

[5] Pala, N., and Cicekli, I. "Turkish Keyphrase Extraction Using KEA", in *Proceedings of 22nd International Symposium on Computer and Information Sciences*, pp. 193-197, Ankara, Turkey, 2007.

[6] Kalaycılar, F., and Cicekli, I., "TurKeyX: Turkish Keyphrase Extractor", in *23rd of the International Symposium on Computer and Information Sciences*, Istanbul, Turkey, 2008.

[7] Villalon, J., and Calvo, R.A., "Concept Extraction from Student Essays, Towards Concept Map Mining", in *Proceedings of the 9th IEEE International Conference on Advanced Learning Technologies*, pp. 221-225. IEEE Computer Society, Washington DC, USA, 2009.

[8] Witten, I.H., Paynter, G.W., Frank, E., Gutwin, C., and Nevill-Manning, C. G., "KEA: Practical Automatic Keyphrase Extraction", *Computer Science Working Paper 00/05*, Hamilton, New Zealand: University of Waikato, 2000.

[9] SPSS Inc. "Gaining Full Value from SPSS Text Analysis for Surveys", Technical Report, GVSTWP-1008, 2008.

[10] Moens, M.F., and Angheluta, R. "Concept Extraction from Legal Cases: The Use of a Statistic of Coincidence", in *Proceedings of the 9th International Conference of Artificial Intelligence and Law*, pp. 142-146, New York, NY, USA: ACM, 2008.

[11] Bing, J., "Performance of Legal Text Retrieval Systems: The Curse of Boole", *Law Library Journal*, vol. 79:187-202, 1987.

[12] Rissland, E. L., Skalak, D. B., and Friedman, M.T., "Bankxx: Supporting legal arguments through heuristic retrieval", *Artificial Intelligence and Law*, 4 (1):1-71, 1996.

[13] Wang, J., Peng, H., and Hu, J., "Automatic Keyphrase Extraction from Document Using Neural Network", *Advances in Machine Learning and Cybernetics*, Berlin, Heidelberg, Springer-Verlag, LNAI 3930:633-641, 2006.

[14] Turney, P., "Learning to Extract Keyphrases from Text", Technical Report, ERB-1057, National Research Council, Institute for Information Technology, Canada, 1999.

[15] Rohini, U., and Ambati, V. "Extracting Keyphrases from books using language modeling approaches", in *Proceedings of the 3rd International Conference on Universal Digital Library*, Pittsburgh, USA, 2007.

[16] Provalis Research, "WordStat v6.0 Content Analysis and Text Mining", Help File, Montreal, Canada, 2009.

[17] Sak, H., Güngör, T., and Saraçlar, M., "Turkish Language Resources: Morphological Parser, Morphological Disambiguator and Web Corpus", in *Proceedings* of *6th International Conference on Natural Language Processing*, LNCS 5221:417-427. Springer, 2008.

[18] Rakotomalala, R., "TANAGRA: A Free Software for Research and Academic Purposes", in *Proceedings of European Grid Conference*, FRNTI-E-3, vol. 2, pp. 697-702, 2005.

[19] Journal of the Faculty of Engineering and Architecture of Gazi University, 20(1-3), 21(1-4), 2006.

[20] Alpaydın, E., *Introduction to Machine Learning, 2e*, London, England: The MIT Press, 2010.

On the Optical Character Recognition and Machine Translation Technology in Arabic: Problems and Solutions

Prof. Oleg Redkin, Dr. Olga Bernikova
Faculty of Asian and African Studies, St. Petersburg State University, St. Petersburg, Russia

Abstract - *The report addresses the basic problems of the Arabic language formalization based on analysis of linguistic errors in software products. Reviewing the principles of modern information systems operation the authors come to the conclusion that the existing methods of the Arabic formalization allow to note a shift towards the technological aspects of the linguistic processing of facts, however, the quality of applied linguistic components still remains poor. Possibilities for the application of traditional recognition algorithms for Arabic are still uncertain in spite of a significant number of theoretical and practical results in the field of computational linguistics. There are several problems which are due to be solved in relation to the processing of the Arabic text. These issues may be divided into those related to Optical Recognition of the Written Text (OCR), Word Processing (WP) and building of the content of the dictionaries, Machine Translation (MT).*

Keywords: NLP, Arabic, OCR, machine translation.

1 Introduction

Modern tendencies of developing in the information-oriented society can be defined by the deep interaction of the international industrial, technical, economic, sociological, scientific and informational aspects. Globalization in every aspect of everyday life is primarily performed through language, which is the main tool of communication. Expansion of cross-cultural interaction on different levels calls for the necessity to eliminate the so-called "language barrier"; this can be achieved through the application of the machine translation systems and improvement of the multilingual search engine software.

Recently, the progress in development of quantity and quality of the software applications for processing of linguistic material in Arabic became evident. However the available linguistic software products still abound in errors and obvious mistakes. This largely happens due to the intention to shift the focus towards the creation of the technological solutions for the universal processing of linguistic material, while the quality of the linguistic background tends to be left in a poor condition.

2 Historical background

The commencement of the language formalization process, aimed at establishing of the automatic parsing of linguistic material dates back to early 60's[1]. However just in 1981 John McCarthy generalized the theory to deal with the conjugational system of Arabic, on the basis of an autosegmental account of vowel and consonant slots on a central timing tier (A CV skeleton analysis of Arabic root-and-pattern morphology) [2].

The first two-level morphological analysis was carried out in 1983 by K. Koskenniemi. At that time he presented a new linguistic, computationally implemented model for morphological analysis and synthesis. It was general in the sense that the same language independent algorithm and the same computer program can operate on a wide range of languages, including highly inflected ones such as Finnish, Russian or Sanskrit. The model was based on a lexicon that defines the word roots, inflectional morphemes and certain nonphonological alternation patterns, and on a set of parallel rules that define phonologically oriented phenomena. The rules are implemented as parallel finite state automata, and the same description can be run both in the producing and in the analyzing direction[3]. His system was based on the prefix-suffix method of word formation and a simple sequence of morphemes. The changes of the internal structure of the root, as well as the conversion of weak roots were not considered at that time.

In 1996, Xerox Research Center has implemented a system using a combination of automated algorithms for root structures and derivational patterns. This technology is proved to be very effective for the linguistic data processing, and now it is used in NooJ - a linguistic environment, processing the texts of a few million units in real time. It includes tools for compiling, testing and supports the bulk of lexical resources, as well as morphological and syntactic grammars[4]. However, our analysis of the NooJ operation shows that the proposed description of the principles of the Arabic language formalization is not comprehensive and, as a result, does not provide the sufficient accuracy of data processing. In

[1]Bar-Hillel, Y., "The Present Status of Automatic Translations of Languages"; Advances in Computers, Vol.1, 91-163, 2006.
[2]McCarthy John J. "A prosodic theory of nonconcatenative morphology"; Linguistic Inquiry, Vol. 12, 373-418, 1981.
[3]Koskenniemi, Kimmo, "Two-level Morphology: A General Computational Model for Word-Form Recognition and Production, Publications", University of Helsinki, Department of General Linguistics, 1983. P.160.
[4] http://www.nooj4nlp.net/pages/nooj.html.

particular, insufficient quantity of verbal derivational algorithms was noted along with the lack of the complex solutions for the root definition based on various patterns of broken plurals, distinctions in the final hamza writing when joining enclitic, etc. Thus, the paper "Standard Arabic Formalization and Linguistic Platform for its Analysis" by Slim Mesfar, describes NooJ as «system that uses finite state technology to parse vowelled texts, as well as partially and not vowelled ones. It is based on large-coverage morphological grammars covering all grammatical rules»[5]. The author highlights that for Arabic, as a highly inflectional and derivational language, they had to define three new operators such as the <T> operator that checks if the last consonant within a noun is a " ة" (T - Teh marbouta) to replace it with a " ت" (t –Teh maftouha) in some inflexional or derivational descriptions:

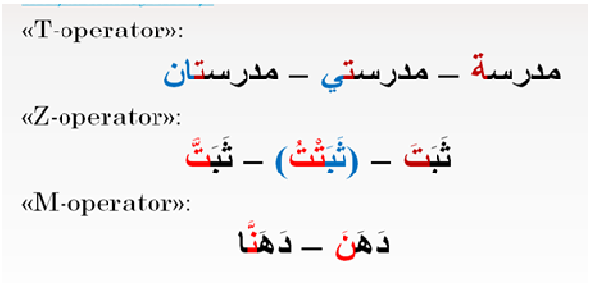


Figure 1. Special "operators" for Arabic in accordance with "Standard Arabic Formalization and Linguistic Platform for its Analysis" by Slim Mesfar.

Noting that the system contains all grammatical rules the author misses a number of grammatical features of the Arabic language. For example, there are no "operators" for broken plurals as there is no distinction in types of plural patterns, depending on the proper part of speech:

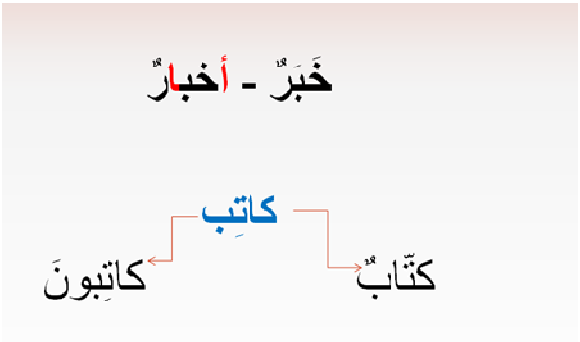


Figure 2. Distinction in plural patterns, depending on the part of speech and the example of broken plural pattern.

Besides that, Slim Mesfar notes that "by inflectional description it was composed the set of all possible transformations which allow us to obtain, from a lexical entry, all inflected forms. These inflexional descriptions include the mood (indicative, subjunctive, jussive or imperative), the voice (active or passive), the gender (masculine or feminine), the number (singular, plural or dual) and the person (first, second or third). On average, there are 122 inflected forms per lexical entry"[6]. According to our research there can be up to 400 forms and, taking into consideration the possibility of enclitics joining for the transitive verbs, up to 780 models.

Our findings in this area are based on experience in the development of the morphological analyzer which has the following characteristics:

- More than 60000 rules for verbs.
- More than 7500 for names.
- The rules do cover all kinds of roots.
- Lexical content includes 20000 units.
- All distinctions in *hamza* and *tashdeed* writing are taking into consideration.

3 Problems

The recent developments in the field of the Information and Communication Technologies open new horizons and perspectives for scholars engaged in the field of linguistic analysis and linguistic data processing. It covers a wide range of areas and allows to create new programs based on the Arabic script such as data bases, automatic translation and automatic Arabic text recognition, search engines in the Internet, teaching tools, and, finally, Arabic-Arabic and Arabic-multilingual dictionaries.

Along with the advantages brought by the common development in the field of Arabic software there is a number of difficulties. One of the most important of them is the proceeding of the Arabic text. In order to cope with these challenges we have to develop new principles and ways, which aim to solve the problem of the Arabic language text formalization.

There are several problems which are due to be solved in relation to the processing of the Arabic text. These issues may be divided into those related to Optical Recognition of the Written Text (OCR), Word Processing (WP) and building of the content of the dictionaries, Machine Translation (MT).
Unlike Latin or Cyrillic scripts, the Arabic one brings along many peculiarities, which cause many difficulties the researches confront with.

Such difficulties are represented by a wide variety of Arabic fonts (the major are kuufii, deewaanii, ruq'a, naskh, thulth, etc.) as well as by elements of individual letters, the various forms of writing of several letters depending on their

[5] Mesfar S. "Standard Arabic formalization and linguistic platform for its analysis"; Proceedings of the Conference The Challenge of Arabic for NLP/MT, London, England, 2006. P.84.

[6] Mesfar S. "Standard Arabic formalization and linguistic platform for its analysis"; Proceedings of the Conference The Challenge of Arabic for NLP/MT, London, England, 2006. P. 86.

position in the word, and ligatures, symbols for germination and vowel signs (diacritics), which are spelled sporadically.
Besides their graphic representation, Arabic letters are distinguished by dots. These dots may be placed above or below letter characters, in hand-written texts and manuscripts the precise location of these dots may vary.
Besides that, there are special characters used in local dialects (i.e. ڤ , پ , ڽ , ڭ , چ), or characters designated for sounds which are not typical to Arabic (ڤ , چ , پ , etc.). should not only be taken into consideration but considered as vital elements.

Along with the vast variety of fonts and handwritings, as well as the specifity of realization of the Arabic letters due to their position in the word, there is a big number of possible written variants of almost every Arabic word. These variants depend not only on the grammatical categories of definite / indefinite article, case (nominative, genitive, accusative), number, but on a written realization of some graphic elements, for example, dots under "yā' - ي ", hamza ء, madda آ, waşla ٱ, tashdeed ّ, etc., i.e. the elements which have a 'facultative' character in modern Arabic texts, especially in printed ones.

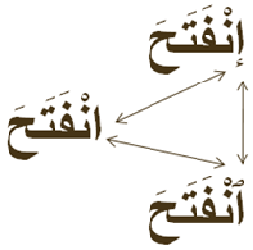


Figure 3. Examples of differences in hamza ء writing.

Even the very preliminary analysis of the Arabic text reveals the difficulties of the automatic morphological classification. The difficulties in formalizing of the Arabic language are due to the vast variety of the possible written representations of words and word forms. So, the number of the word variations derived from one stem is very significant.

Thus for maşdars, participles, and adjectives and nomina loci et temporis, and other nouns the great variety of forms is resulted by various prefixes and suffixes, inseparable prepositions, inseparable conjunctions, terminations of declension and broken plurals, etc. The variety of forms is also caused by the infixes of the eighth stem and different morphological models of the broken plurals.

Adding to this paradigm the possible variants of the written forms (with or without hamza ء, tashdeed ـّـ , waşla ٱ and madda آ), and not taking into consideration the vocalization structures, which actually reduces the total number of the forms, the list of written variants is very vast. For example, there are around 200 written variants of the word باب 'door'.

As for the paradigm of the verbal forms, in comparison with the noun ones the total number is much higher due to the more complicated and developed system of the verb conjugation in Arabic. Adding to this three grammatical categories of numbers – singular, dual and plural, active and passive voices, two genders, verbal forms in the first, second, and third persons (in dual – the second and the third persons), and, finally, five moods, one can imagine the huge spectrum of the verbal paradigm in Arabic.

Thus, the number of the forms only in the active voice of the first form (perfect and imperfect, five moods) is as high as 76. As for the graphical realization the optional representations are possible (with or without hamza, tashdeed, waşla and madda). Besides that, the inseparable particles increase the number of possible written forms of the subjunctive, jussive, energetic and imperative moods due to the number of the particles. The vocalization increases the total number of the derivates and may double it.

The total number of the verbal derivates in all possible stems may include several thousands and, with optional variants, the total number is up to 10000 forms. For example, the verb مرّ 'to move' has more than 12000 possible written derivates, including all possible stems and optional written variants (without vocalization)

Thus, the search engines based on the alphabetic principle such as in Indo-European languages, in Arabic in many cases prove to be ineffective.

4 Optical Character Recognition (OCR)

All the mentioned above incredibly multiplies the number of the written variants in Arabic and, as a consequence, difficulties of the Arabic OCR. The lack of precise standards of written Arabic along with the multiplicity of the grammar forms lead to complex problems in classification of the images of separate characters and words. It is necessary to develop new attitudes and solutions which would be valid within the framework of multivariate and multicriteria analysis, and which will be adapted for the peculiarities of a certain text.

Traditional algorithms of OCR for Arabic still remain insufficiently developed in spite of a big number of theoretical and practical achievements in the field of computational linguistics. The character recognition problem of Arabic text is more complex than that one in terms of texts based on Roman or Cyrillic alphabets.

Effectiveness of existing character recognition systems for Arabic to a great extent depends on the structure of a specific text and may vary within a very significant range.

In many cases the search engines based on the alphabetic principle, which proved to be effective enough for the Indo-European languages, are ineffective for the Arabic language.
Meanwhile, the systematic description and automatic analysis of the data base of the Arabic texts would provide the creation of the empiric principles of character recognition and search engines in Arabic. Researches of this kind have been

successfully carried out for several years at the Arabic Chair of St. Petersburg State University, Russia.

The results of OCR besides the basic characteristics of computer soft- and hardware depend on the quality of the text, the font size and its type, quality of paper, type of printing, printer ink, colors etc.

Figure 4. The sample of the vertical text segmentation in accordance with "Automatic Recognition of the Printed Arabic Texts", Shalymov D.S.[7]

The OCR of handwritten texts, in which the variation of Arabic letters is higher, is even more difficult. For instance, there is possibility of the displacement of the diacritics (dots) with respect to key elements of alphabetic characters in handwritten texts. Since the Arabic alphabet has 18 letters which differ by the number of dots and their positions (above or below the character), the dots may be considered as one of the most relevant elements in the OCR process. The question is twofold: 1) how to define the possible spectrum of the variety of the key elements of certain characters and how to define the borders of the of the position options of the elements against each other? In other words: what is the permissible scale of the transformation of characters in Arabic and dispersion of the elements which compose the characters? And how far from the main character element the dots can be located and what are the possible shapes of the dots that allow the optical recognition of the word?

Since the perception and the recognition of a certain character and combination of characters is a cognitive process based on analysis the graphic environment within a word, and, finally, the use of the graphic, morphologic and semantic contexts of the sentence or the entire text.

The cognitive process of the analysis can be phased (recognition of a character, recognition of a character within the context of a word, recognition of a character within the context of a word -(sentence) text), or take place simultaneously.

In computer analysis each stage requires real-time data processing. The amount of the data depends on the level of analysis – i.e. analysis of the optical data on the initial stage, optical and morphologic and semantic data on the word level, optical-morphologic-semantic syntax data within the context of a sentence, etc.

The huge amount of data to be analyzed demands new attitudes and approaches in building mathematical model and simulators of this process.

[7] Shalymov D.S. "Automatic Recognition of the Printed Arabic Texts"; Stochastic Optimization in Computer Science, Vol.3, 2009. P.125.

The solution may be found on the basis of the technology of simultaneous perturbation stochastic approximation algorithm (SPSA) developed by Prof. O.Granichin [6], [7] and Dr. D.Shalymov [18] - the methods which can be applied not only to Arabic but to for other languages as well.

4 Machine Translation

The thorough user-oriented study of the automatic translation systems available today and operating online with a free access has demonstrated the significant growth in the number of translation resources of Arabic. Initial testing of these translation platforms allowed us to determine several systems that carry out the most accurate translation. As the primary resource platform we will select http://translate.google.com. This choice has been created due to several factors. First, we wanted to analyze the curve of the system operation improvement over the past 5-6 years. Since the materials regarding errors in the system (http://translate.google.com operation) from 2006 became available to us, it was interesting to review the translation of the same model in a diachronic and analyze the process of the linguistic components improvement. On the other hand the resource like http://translate.google.com is one of most popular tools for on-line automatic translation, addressed by thousands of Internet users daily.

The following resources were selected as the objects for selective comparison of incorrect translations: http://www.apptek.com, http://www.translation.babylon.com, http://www.systranet.com.

This study aims to identify the most typical linguistic misinterpretations and errors that occur in the leading systems of automatic translation from Arabic into English and Russian. The role of English in this context is considered through the prism of its frequent use as an intermediate language ("hub-language").

The analysis of the quality of automatic translation from Arabic into English and Russian leads to the following conclusions, based on typical errors:

1. Earlier we have already noted that the allowable variation in writing hamza creates difficulties in the formalization of the Arabic language which is certainly reflected in the work of computer-aided translation.

Furthermore, a series of examples demonstrates the fact that the system cannot find the grapheme و (waw), acting as an "and" if the next word starts with أ:

وأكد ياكونين الذي تحدث عن نتائج زيارته الأخيرة الى الولايات المتحدة أن شركته مهتمة بالتجربة الأمريكية في هذا المجال

«The Yakunin, who spoke about his recent visit to the United States that his company is interested in the American experience in this area».

«Якунин, который рассказал о своей недавней поездке в Соединенные Штаты, что его компания заинтересована в американский опыт в этой области».

This misinterpretation takes place due to an error of the system (Google translation) algorithms which are only partially represented by a combination of words containing the initial Hamza. It is interesting to note that when testing the other above mentioned translation systems the given problem is mostly preserved.

Thus, the modern systems of automatic translation should enhance the quality of its operations by means of the creation of the rules for all possible variants of the hamza writing.

2. Another complex problem of the machine translation systems corresponds to the morphological system of Arabic.

The two important aspects are necessary to be identified here.

Firstly, the correctness of the morphological values transfer in the Arabic language is largely dependent on the specifics of its writing system. Thus the absence of vocalizations in the letter leads to the disappearing of some markers of inflectional patterns, which in its turn leads to emerging of errors in the translation. Secondly,, the morphological paradigm of the Arabic language is only partially presented in the modern computer-aided translation.

Thus the example below demonstrates that the system does not translate the verb in the feminine, subjunctive mood. While the translation of the same word in the masculine is given.

فقال الزوج : لو لم تفتح فمها ما وصلت إلى هذا
المصير، لذلك من الأفضل لك أن *تلتزمي* الصمت

«Муж сказал: не открыть рот, если вы к этому судьба, так что это лучшее, что вы *Tltzme* молчания».

The other example is the incorrect translation of the future tense (the suffix wasn't considered):

وأضاف أن الطرفين *سيوقعان* قريبا على مذكرة تفاهم بهذا الشأن تضم أيضا التعاون في إنشاء مراكز لإجراء اختبار الابتكارات الجديدة في مجال صنع القطارات

«Обе стороны *подписали (Past)* меморандум о взаимопонимании в связи с этим также включает в себя сотрудничество в области создания центров для испытания новых инноваций в производство поездов».

The decision in this context would be the integration into the system of models based on the synthesis of the morphological models and the semantics of the used tokens.

3. Another problem of computer-aided translation is related to the principles of cross-language correspondence. We are talking about the fact that the information of the original text is firstly expressed in the neutral intermediate language, and then translated into the target language. The situation of this kind occurs in translation from Russian to Arabic and vice versa. It is known that typologically the Arabic and Russian languages are synthetic, while the English is analytical. Therefore, the use of English as the intermediary language brings errors to the grammatical meanings of words. Thus, the process of algorithms creating for correspondence of the morphological patterns in Russian and Arabic would help to enhance the quality of translation.

ولعل من أبرز نتائج التعاون، ازدياد عدد *الطلبة السعوديين* في الولايات المتحدة الأمريكية من حوالي ثلاثة آلاف طالب في السنوات القليلة الماضية، إلى نحو خمسة وعشرين الف طالب حاليا

«Perhaps the most important results of cooperation, the increasing number of *Saudi students* in the United States of about three thousand students in the past few years, to about twenty-five thousand students at present».

«Пожалуй, наиболее важных результатов сотрудничества, увеличение числа *Саудовской студентов* в Соединенных Штатах около 3000 студентов в последние несколько лет около 25 000 студентов в настоящее время».

It is necessary to note that sometimes we find mistakes in translation by means of so called "neutral language" (i.e. English in this case):

لن نخرج من مصر

«*Will not come out* of Egypt».

«*Не вышел* из Египта».

Thus, to ensure high-quality linguistic software perfection the structural formalization of the language must be flexible and comprehensive. The optimal development of linguistic software products requires application of comprehensive solution that includes the dictionary entries forming on the basis of frequency index, creation of the algorithms inflectional patterns of Arabic and also taking into account the semantic aspects.

«Perhaps the most important results of cooperation, the increasing number of Saudi students in the United States of about three thousand students in the past few years, to about twenty-five thousand students at present».

5 Conclusions

5.1 Considering the complexity and multifaceted character of the problem of the Arabic text formalization and OC recognition, the efforts of the specialists of different specializations - linguists (selection of the linguistic content, problems of the linguistic data parsing) and applied mathematicians (image and voice recognition, filtering, adaptive systems and software design) should be joined.

5.2 The effectiveness of the search engine should be based on automatic methods of the lexical material formalization, i.e. the definition the initial word and the root morpheme from the variety of written forms. Since in Arabic the root morphemes play the most important role as the 'holder' of the basic meanings and ideas, the lemmatization is also important for building of electronic dictionaries and automatic translation.

5.3 There is a big number of written variants of Arabic verbs and nouns which depends on grammatical categories of definite / indefinite, case (nominative, genitive, accusative), number and written realization (absence of the realization) of some graphic elements: such as dots under "yā' - ي ", hamza,

madda, waşla, tashdeed, etc., i.e. the elements which have facultative character in modern Arabic texts, especially in printed ones. So the number of the written variants is very big. The lowest number have separable prepositions, the highest variety of the written forms have inseparable particles and prepositions. In other words, those particles may be written with the majority of the noun or verbal forms.

Thus, the initial task is to develop automatic methods of the formalization of the lexical material and derive from the variety of word forms the vocabulary word form, and, finally, the root morpheme.

It will enable to create the linguistic content of the electronic dictionary on the basis of written and electronic Arabic texts, and the main criteria here is the frequency of the word entering into the text. It allows us to include the most frequent words into the linguistic content of the dictionary, and optimize the content of the dictionary.

The electronic dictionary based on the principle of the frequency code is very important, especially on the initial stages of studying Arabic.

As a result computerized methods of the definition of the frequency code of the Arabic words have been developed (the number the word entrances in the entire text on the basis of the analysis of about one million Arabic words included into vast variety of texts).

Table 1. Words in random with frequency code on the basis of analysis of Arabic written texts (about one million words).

شمس	1365
هذه	1365
إن	1275
عام	1274
ل	1271
مع	1175
ذلك	1096
علي	1089

Except for the limited number of dictionaries, the authors do not explain or stipulate the principles of the formation of the linguistic content and reflect authors' personal affiliations and criterion.

On the contrary, the words frequency code was the main criterion in the process of building the lexical content of the new Arabic-Russian dictionary.

Thus, the basic principles of the description of Arabic typical structures were developed along with the electronic Russian-Arabic and Arabic-Russian dictionaries.[8]

The principles of Arabic text processing were developed in cooperation with the team of mathematicians from St. Petersburg State University, among them O.Granichin and D Shalymov, who developed the implementations of various randomized algorithms of the stochastic multidimensional optimization such as SPSA (Simultaneous Perturbation Stochastic Approximation).[9] These principles may be used for the adaptive classification of the images and signals in the framework of uncertainty.

5.4 The research of the existing methods of the linguistic analysis, along with the positive experience in compilation of a set of morphological paradigms and dictionaries of Arabic, allows to develop a language model, which would be able to become the background for the development and improvement of machine translation technology, search engines, as well as the creation if linguistic software, and, on the other hand, it would contain the maximum possible amount of information for linguistic research. The proposed model focuses on the interaction between the morphological base, thematically tagged dictionary and text corpus. Each of these elements carries its functional load, however, only their complex integration could contribute to the improvement of technological solutions for Arabic.

5 References

[1] Bar-Hillel, Y., "The Present Status of Automatic Translations of Languages"; Advances in Computers, Vol.1, 91-163, 2006.

[2] Bernikova O.A. "The Arabic Language as a Tool for Informational Systems Providing (in Russian)"; Proceedings of the XXVI International Conference on Source Studies and Historiography of Asia and Africa "Modernization and tradition", St.Petersburg, Russia, 346-347, April 2011.

[3] Bernikova O.A. "The Arabic Grammar in Tables and Diagrams (in Russian)"; St.Petersburg, Russia, 2010.

[4] Bernikova O.A. "Problems of the Arabic Language Formalization (in Russian)"; Proceedings of the Conference "Oriental Studies and Dialog of Civilizations", St.Petersburg, Russia, 371-372, April 2009.

[5] Bernikova O.A. "Oriental Languages Distance Learning System (in Russian)"; Saint-Petersburg State University, Vol.17, 17-19, 2007.

[6] Granichin O.N., Vakhitov A.T., Gurevich L.S. "Algorithm for Stochastic Approximation with Trial Input

[8] Redkin O.I., Bernikova O.A., Shalymov D.S. "Software for Arabic text e-learning, translation and recognition"; St.Petersburg, Russia, April 2007. Certificate of the Official Software Registration No. 2007611711 dated 23.04.2007 «Software for Arabic Text e-Learning, Translation and Recognition», Russian Federation IP Agency.

[9] Vakhitov A.T., Granichin O.N., Sysoev S.S. Accuracy of Randomized Algorithm for Stochastic Approximation // Automation and Remote Control, 2006, v.67, No. 4, pp.589-597.

Perturbation in the Nonstationary Problem of Optimization"; Automation and Remote Control, Vol. 70, no. 11, 1827-1835, 2009.

[7] Granichin O.N., Hieu L.T. "Using Application of Statistics for Word Extraction from Vietnamese documents" ; Vestnik Sankt-Petersb. Univ. App. Math, no. 3, pp. 162-170, 2009.

[8] Granichin O.N., Vakhitov A.T., Gurevich L.S., "Automation and Remote Control", 2009.

[9] Granichin O.N., Polyak B.T. "Randomized Algorithms of an Estimation and Optimization Under Almost Arbitrary Noises". Nauka, Moscow, 2003.

[10] Izwaini S., "Problems of Arabic Machine Translation"; Proceedings of the International Conference on the Challenges of Arabic Machine Translation and Natural Language Processing, British Computer Society, London, 118-148, 23 October 2006.

[11] Mesfar S. "Standard Arabic Formalization and Linguistic Platform for its Analysis"; Proceedings of the Conference The Challenge of Arabic for NLP/MT, London, England, 84-95, 2006.

[12] McCarthy John J. "A Prosodic Theory of Nonconcatenative Morphology"; Linguistic Inquiry, Vol. 12, 373-418, 1981.

[13] Koskenniemi, Kimmo, "Two-level Morphology: A General Computational Model for Word-Form Recognition and Production, Publications", University of Helsinki, Department of General Linguistics, 1983.

[14] Redkin O.I. "Classical School of Oriental and Digital Technologies - New Opportunities (in Russian)"; Proceedings of the XXVI International Conference on Source Studies and Historiography of Asia and Africa "Modernization and tradition", St.Petersburg, Russia, 367-368, April 2011.

[15] Redkin O.I. "Formation of the Thesaurus of Arabic on the Basis of Frequency (in Russian)"; Proceedings of the Conference "Oriental Studies and Dialog of Civilizations", St.Petersburg, Russia, 379-380, April 2009.

[16] Redkin O.I. , Bernikova O.A., "Applying of technical equipment in teaching of the Arabic language"; Asian and African Studies, St.Petersburg, 124, 2006.

[17] Redkin O.I., Bernikova O.A., Shalymov D.S. "Software for Arabic text e-learning, translation and recognition"; St.Petersburg, Russia, April 2007.

[18] Shalymov D S. "Continuous Speech Recognition Using Simultaneous Perturbation Stochastic Approximation Algorithm"; Vestnik St.Petersburg State University, Vol.3, 171-181, 2009.

[19] Shalymov D S. "Automatic Recognition of the Printed Arabic Texts"; Stochastic Optimization in Computer Science, Vol.3, 124-137, 2009.

Semi-automatic verb-driven lexicon acquisition enhancer

Max Petrenko
Princess Ekaterina R. Dashkova Moscow Humanities Institute
and RiverGlass, Inc.
Moscow, Russia, and Champaign, IL, USA
maxpetrenko@alumni.purdue.edu

Abstract – *the paper discusses ways of enhancing lexical acquisition within Ontological Semantic Technology without the detriment to the grain size of lexical coverage. The semi-automatic verb driven enhancer is introduced which derives nominal and adjectival senses from manually acquired verb senses by converting their case role fillers into complete sense templates. The declarative and procedural components of the enhancer are described, and its functionality is demonstrated.*

Keywords: computational lexicon, lexicon acquisition, language engineering, ontological semantics

1 Introduction

The need and rationale for developing an incremental utility for enhancing lexicon acquisition emerged out of several years of semi-automatic acquisition of the English lexicon within the framework of Ontological Semantic Technology (OST) – an already well-established and fully functional system featuring a full-fledged methodology and toolbox for knowledge acquisition. While the automation of lexicon acquisition in OST is already achieved through format consistency check in lexical senses and ontological fillers validity check in semantic structures, further possibilities have become available, mostly prompted by the richness of the increasingly robust ontology. From the enhancer's adherence to the OST framework, two premises follow. First, the declarative and procedural components of the enhancer fully reflect the premises, methodology of lexicon acquisition postulated in [3] and further discussed in [6]. Second, the introduced utility aims at enhancing, rather than supplanting, the efforts of a human acquirer, and is meant to further address the common charges of time-inefficiency of acquisition efforts and the incompleteness and non-uniformity of resources traditionally leveled against meaning-driven human-mediated lexicon acquisition [4]. The latter premise also implies that the incremental application of the enhancer does not warrant its systemic comparison to large-scale fully automatic systems like [5].

The paper will briefly outline the framework of OST, discussing pertinent aspects and lexicon acquisition strategies and techniques discussed in [3] and [6]. The semi-automatic verb-driven enhancer will then be introduced in detail. The enhancer's functionality and efficiency will then be illustrated with a detailed step-by-step description of senses extraction from a verb sense.

1.1 Knowledge resources of OST

As argued, illustrated and practically demonstrated in an increasingly growing number of publications [1, 7, 8, and 3], OST is predicated on the premise that scalable computation of natural language text requires a robust, descriptively rich and reusable knowledge resource platform on the basis of which natural language meaning could be accessed directly rather than being guessed via collocational or syntactic dependencies [7].

The knowledge resource platform comprises a language-independent ontology – a lattice of concepts defined through a rich network of properties – and a language-specific lexicon, capturing word senses including orthographic, syntactic alternations, incorporating optional prepositional or phrasal adjuncts, and representing non-compositional units.

The architecture, axiomatic description and concept structure has been extensively discussed in [4, 8]. Stemming from the top-level OBJECT-EVENT-PROPERTY branching, OBJECT and EVENT concepts are defined via an extensive set of PROPERTY concepts in the (concept(property(facet(filler)))) format. Inheritance and transitivity relations are enforced. Filler preference is captured via facets SEM and DEFAULT, and numeric value ranges of ATTRIBUTEs can be represented via RELAXABLE-TO.

Lexicon entries relate natural language syntax and semantics by co-indexing syntactic positions (subject,

direct object, etc.) with semantic property fillers (CASE-ROLE's for EVENT senses, RELATION's for OBJECT senses, and DOMAIN/RANGE for ATTRIBUTE senses). A lexical entry template is shown below:

```
(head-entry
    (sense-1, 2, 3…
        (cat(n/v/adj/pro/prep))
        (synonyms "")
        (anno
            (def "")
            (comments "acquisition time stamp or other
notes")
            (ex "")
        )
        (syn-struc(
            (root($var0))(cat(n/v/adj/pro/prep))
            (subject/object((root($var#))(cat(np/vp/s))))
        )
        (sem-struc
            (root-concept
                (property(value(^$var#
                (restricted-to(default/sem(concept)))))))
        )
    )
)
```

During computation, the text processor will derive a text meaning representation (TMR) of a sentence by mapping syntactic positions onto semantic roles, in the order stipulated by the syntactic structure ("syn-struc") of the sense, where $var0 stands for the word sense per se, and all additional variables indicate adjacent optional or obligatory arguments. Order variation is accommodated through additional syntactic structures. The semantic structure ("sem-struc") defines the meaning of the sense by anchoring it in the most representative concept (i.e. root-concept) adding properties where necessary. Restriction of property filler variables is often required because anchor concepts may be too generic to capture a particular sense. An example sense of "ax-v1" is listed below:

```
(ax-v1
        (cat(v))
        (anno(def "to destroy or break with an
ax")(comments "")
    (ex "he axed the door to break out of the burning
house")(synonyms "")
        (syn-struc(
        (subject((root($var1))(cat(np))))
                                (root($var0))(cat(v))
    (directobject((root($var2))(cat(np))))))
    (sem-struc(destroy(agent(value(^$var1(restricted-
to(human)))))(instrument(sem(ax)))
    (theme(value(^$var2(restricted-to(sem(plant
artifact)))))))))
```

Since DESTROY and its case roles of AGENT, INSTRUMENT and THEME are too general to capture the meaning of "ax-v1", case role filers are "tightened": from ANIMAL to HUMAN (for the AGENT case role), from TOOL to AX (for the INSTRUMENT case role), and from PHYSICAL-OBJECT to PLANT ARTIFACT (for the THEME case role).

1.2 Lexicon acquisition in Ontological Semantics

In OST, lexicon acquisition is guided by the following premises [see 3 for a detailed discussion]:

1) Lexicon coverage must be complete, i.e. represent all senses, for a target domain;
2) The breadth and depth (i.e. "grain size") of lexicon acquisition is determined by application objectives;
3) Maintaining a determined grain size is crucial for text meaning representation and overrides considerations of time efficiency;
4) Automatic (i.e. "runtime") rule-based sense acquisition has no decisive advantage over semi-automatic rule-based sense acquisition in OST because the generative capacity of acquisition rules is unaffected by whether they are triggered at runtime or acquisition time.
5) Human involvement in OST lexicon acquisition is instrumental in aspects where semantic competence is needed: selecting an appropriate head concept, determining the optionality of syntactic elements, defining the number of senses per entry, reducing polysemy and post-acquisition testing.
6) The automation of lexicon acquisition in OST should be incremental, i.e. used sparingly so as to avoid supplanting or impeding human involvement in semantic aspects outlined in 5.

Rule-based and propagation-based strategies of lexicon acquisition have been described in [3], and ontology-driven, parser-driven or domain-driven acquisition techniques outlined in [6].

2 Enhancer description

The section introduces the declarative (goals, resource components, input/output specifications) and procedural (rules and derivation phases) components of the semi-automatic verb-driven lexicon acquisition enhancer.

2.1 Declarative component of the enhancer

The enhancer generates nominal and adjectival senses from a manually acquired verb sense. The generation involves "unpacking" case role fillers (obtainable from case role variable restrictions or ontological case role fillers) of the verb sense into respective noun senses, or syn-struc and sem-struc modifications of the verb sense in order to derive deverbal nouns or adjectives. Inverse property calculation is crucial for the derivation. Once a complete lexical sense (including the annotation, syn-struc and sem-struc fields) has been created, an appropriate name is looked up and, upon the approval of a trained acquirer, assigned to the sense. Once a sense has been named and confirmed by the acquirer, an appropriate synonym list is automatically retrieved for the given name, and the sense's complete template is propagated throughout the synonym list. The procedure is semi-automatic in that the enhancer will first attempt to derive a sense automatically and will request acquirer's approval at every phase of the process.

The enhancer builds on the already existing lexicon acquisition technology described in [3, 6]. In order to increase the breadth of coverage for the enhancer output, a repository of noun and adjective synonym lists is included as a separate resource component. The synonym lists are formalism-free, i.e. consist of simple words without definitions, notational encoding or ordering preferences:

<house dwelling homestead home shack ...>
<workable doable practical feasible applicable…>

What matters for the enhancer is a paradigmatic lexical set with predetermined semantic similarity, which will receive full interpretation upon the completion of the enhancement procedure.

Two types of verb senses serve as input to the enhancer: verbs with unrestricted case role variables; and verbs with restricted case role variables.

An unrestricted case role variable implies that the case role filler is specified in the ontology and no tightening is necessary for the sense. Consider the sem-struc of "travel-v1":

```
(travel-v1
    (cat(v))
    (anno(def "to move via transport") (ex "He traveled from France to Spain. The ship traveled to South Africa"))
    <…>
     (syn-struc((subject((root($var1))(cat(np))))
                    (root($var0))(cat(v))))
    (sem-struc(travel(agent(value(^$var1)))
                    (instrument(value(^$var1)))))
)
```

Both AGENT and INSTRUMENT case roles are sufficiently constrained for the concept TRAVEL in the ontology:

```
(travel
        (definition (value("to move via transport")))
        (is-a (hier(change-location culture-event)))
        (subclasses(hier(drive)))
        (agent(sem(human)))
        (instrument(sem(vehicle)))
        <…>
)
```

From the enhancer perspective, deriving a sense from an unrestricted case role variable means retrieving the sem-struc head concept from the respective ontological filler of the given case role.

A restricted case role variable indicates that the case role filler specification in the ontology is too generic for the given verb sense and needs tightening. Consider a verb sense of "board":

```
 (board-v1
        (cat(v))
        (anno(def "to move on board of a vehicle")
         (ex "the passengers boarded the plane"))
        (syn-struc((subject((root($var1))(cat(np))))
(root($var0))(cat(v))(directobject((root($var2))(cat(np))))))
        (sem-struc(walk(agent(value(^$var1)))
        (destination(value(^$var2(restricted-
to(sem(vehicle)))))))))
```

No directly appropriate concept for the event of boarding transport exists in the ontology, and the restrictions on the DESTINATION case role for the nearest available concept WALK are too general to capture the meaning of "board-v1", therefore the DESTINATION filler is "tightened" from PHYSICAL-OBJECT to VEHICLE. From the enhancer perspective, a case role variable restriction often contains, other conditions being met, an explicitly defined sem-struc head of the new entry.

The enhancer generates the following types of senses:

a) Case role-based noun senses are derived from case role fillers of the verb and are listed below by respective case roles in the ontology:

- agentival noun (AN): research-v → researcher-n, scholar-n, scientist-n, etc.
- instrumental noun (IN): eat-v → spoon-n, fork-n,
- thematic noun (TN): discard-v → trash-n, garbage-n, junk-n, etc.
- benefactive noun (BN): drive-v → passenger-n
- locative noun (LN) derived from case roles of LOCATION, PATH, SOURCE, DESTINATION: meal-v → restaurant-n; board-v → gangway-n; jump-v → springboard-n, etc.

b) Deverbal noun senses are derived by converting syntactic positions of the verb into optional

prepositional adjuncts without any modifications to the sem-struc: research-v → research-n; drive-v → drive-n, etc.

c) Adjectival senses are normally derived by modifying the syn-struc and sem-struc of the verb: work-v → workable-adj, eat-v → edible-adj

At the sense approval/revision phase, it is important for the acquirer to remember that the enhancer is aimed at generating only those nominal and adjectival senses whose semantics is directly informed by the case role properties used in the sem-struc of the verb. To illustrate, an agentival noun sense "teacher-n" can be derived from "teach-v1". While the AGENT case role of the sem-struc of "teach-v" warrants any noun sense anchored in HUMAN (driver-n, brother-n, criminal-n, etc.), only "teacher-n1" will have its semantics exhaustively encapsulated by the (human(agent-of(default(teach)))) expression. Simply put, any human is capable of teaching, but for "teacher" the quality of being the agent of teaching defines the word's existence. To establish the indispensability of a case role property for the meaning of a sense, a criterion can be applied: if eliminating the case role property from the sem-struc of the sense (e.g. "teacher-n") makes it synonymous with its hypernym (e.g. "human-n"), then the case role property is semantically indispensable (a similar distinction between categorical semantic markers shared by all class members and sense-specific distinguishers delineating a specific sense was drawn in [2]).

2.2 Procedural component of the enhancer

The enhancer involves three procedures. The description below spares the reader many algorithmic details and rather aims to outline the general logic of the cycle. The three procedures are not linearly ordered and may be called recursively.

(1) **Sense derivation**

Case role noun derivation

The general rule involves creating an inverse of the respective case role in the sem-struc of the verb sense, so that (event(case-role(facet(filler)))) → (filler(case-role_inverse(facet(event)))).

For example, an agentival noun sense will have its sem-struc derived by replacing the AGENT case role with AGENT-OF and reversing its domain and range fillers so the original (event(agent(sem/default(filler)))) is rewritten as (filler(agent-of(sem/default(event)))). Other case role noun inverse derivations are listed below:

IN: (filler(instrument-of(sem/default(event))))
TN: (filler(theme-of(sem/default(event))))
BN: (filler(beneficiary-of(sem/default(event))))

LN: { (filler(location-of(sem/default(event)))) (filler(path-of(sem/default(event)))) (filler(source-of(sem/default(event)))) (filler(destination-of(sem/default(event)))) }

If the verb sem-struc contains an unrestricted case role variable, the ontological filler is found under the respective EVENT: the (filler(case-role_inverse(sem/default(event)))) string is searched for in the definition of the respective EVENT case role filler or any of its children. If the search returns a concept, it becomes the head of the sem-struc the derived noun sense. Upon acquirer's approval, the sense is created with the anno and syn-struc fields copied from the noun template and the found concept in the head of the sem-struc. The inverse property is omitted to avoid the reduplication of ontological properties:

```
(...-n1
    (cat(n))
    (synonyms "")
    (anno(def "")(comments "")(ex ""))
    (syn-struc((root($var0))(cat(n))))
    (sem-struc(filler))
)
```

If the verb sem-struc contains a restricted case role variable, the inverse expression is derived from the variable restriction based on the respective case role rule, and the (filler(case-role_inverse(sem/default(event)))) string becomes the sem-struc of the noun entry. A complete noun sense is created and offered for acquirer's approval:

```
(...-n1
    (cat(n))
    (synonyms "")
    (anno(def "")(comments "")(ex ""))
    (syn-struc((root($var0))(cat(n))))
    (sem-struc(filler(case-
    role_inverse(sem/default(event)))))
)
```

Deverbal noun derivation

Deverbal nouns are derived syntactically, i.e. by replacing the syntactic roles of subject and object of the verb syn-struc with optional prepositional adjuncts. The conversion rules are listed below:

For subject:

(subject((root($var1))(cat(np)))) → (pp-adjunct((root(by))(opt(+))(cat(prep))(obj((root($var1))(cat(np))))))

For direct object:

(directobject((root($var2))(cat(adj)))) → (pp-adjunct((root(of))(opt(+))(cat(prep))(obj((root($var2))(cat(np))))))

The sem-struc of the verb sense remains unchanged. After the syn-struc conversion is complete, the sense is created:

```
(...-n1
        (cat(n))(synonyms "")
        (anno(def "an act of __") (ex ""))
        (syn-struc((root($var0))(cat(n))
        (pp-adjunct((root(of))(opt(+))
            (cat(prep))(obj((root($var1))(cat(np))))))
        (pp-adjunct((root(by))(opt(+))
            (cat(prep))(obj((root($var2))(cat(np))))))))
        (sem-struc(copied from the verb sense))
)
```

Adjective derivation

The class of deverbal adjectives is very narrow and restricted only to those modifying likely non-agentival case role fillers of the EVENT in the verb sem-struc. The derivation procedure involves creating a standard adjectival sense whose head sem-struc variable is defined through the inverse property of any case role filler other than agent indicated in the verb sense, and a high probabilistic modality value, e.g. (epistemic(value(0.8))) assigned to the EVENT:

```
 (…-adj
        (cat(adj))
        (anno(def "")(comments "")(ex ""))
        (synonyms "")
        (syn-struc((root($var0))(cat(adj))
                (syn-np((root($var1))(cat(np))))))
        (syn-struc1((syn-np((root($var1))(cat(np))))
                (vp((root(be))(cat(v))))(adj-
        comp((root($var0))(cat(adj))))))
        (sem-struc(^$var1(case-
role_inverse(sem/default(event(epistemic(value(0.8))
))))))
)
```

Thus, while deriving deverbal adjectives, the enhancer will ignore the agent and try and convert any other case role filler, e.g.:

THEME (read-v → readable-adj)
BENEFICIARY (adore-v → adorable-adj)
INSTRUMENT (drive-v → drivable-adj)
PATH (pass-v → passable-adj)

(2) **Name assignment**

The name assignment procedure determines whether the newly created sense skeleton, complete with syntactic and semantic specifications, is lexicalized in English, i.e. has a name and enters a synonym class. While automatic name search steps will first be undertaken, acquirer's approval is often crucial at this phase. In order to establish lexicalization, the system will exploit the property of ontological concept labels which acquirers trained in OST are normally instructed to be conscious of: the possible overlap of the concepts labels with language lexemes. This said, presuming that, while concepts and their labels are language independent, the choice of a concept label may have been motivated by a prototypical lexical token, the system will check whether the label of the sem-struc head concept matches any of the words available in the synonym list repository. If a match is found, the label will be suggested for the sense name, and acquirer's approval will be requested. If no match is found, the acquirer will be requested to assign the name manually.

Manual name assignment is most often required in adjectival sense derivation for two reasons. First, most deverbal adjectives reluctantly form synonymous relations, which reduces the available synonym list repository thus preventing automatic search. Second, while deverbal adjectives of the type above at the first glance have a relatively uniformed morphological structure and seem to easily lend themselves to automation, a detailed look reveals a more complex picture. The suffixation process involves internal variation in the suffix (-able/ible) and the verb stem (e.g. final consonant reduplication in cases like "regrettable") based on the morphophonemic properties of the stem, the surface phonetic representation of the final "e" (e.g. value → valuable, but foresee → foreseeable) – a task whose automation, if possible and justifiable, requires a separate research and development effort. Having the acquirer assign the name manually would seem a more time-efficient strategy in the current implementation.

The acquirer's role at this phase is determining whether the created sense deserves a name, i.e. a word exists in English whose meaning is exhaustively encapsulated in the sem-struc of the created sense; the indispensability criterion as described in 2.1. is also made use of.

(3) **Synonym check**

Synonyms are assigned to the created sense by propagating its template among all members of the synonym list. Acquirer's approval is required to establish the semantic grain size of the coverage, which will also determine the synonym class size.

3 Enhancer functionality

The current section illustrates how the semi-automatic verb-driven lexicon acquisition enhancer generates nominal and adjectival senses from the manually acquired verb sense of "drive-v":

```
(drive-v1
        (cat(v))
        (anno(def "to change location by operating a
        vehicle")
        (ex "he drove his car. he drove across the
city. tanks drove in the city")(synonyms "")
         (syn-struc((subject((root($var1))(cat(np))))
(root($var0))(cat(v))(directobject((root($var2))(opt(+
))(cat(np))))))
        (syn-struc1((subject((root($var2))(cat(np))))
(root($var0))(cat(v))))
        (sem-struc(drive
        (agent(value(^$var1)))
        (instrument(value(^$var2)))))
)
```

Deriving case role nouns from "drive-v"
Agentival nouns

Since the variable ^$var1 on AGENT is unrestricted, ontological fillers of AGENT for DRIVE are checked, and the concept HUMAN is detected. The inverse of (drive(agent(sem(human)))) is derived, and HUMAN and all its children concepts are checked for the expression (agent-of(sem(drive))) in their definition. After the automatic search returns no results, the acquirer is asked to manually add the head concept:

```
(-n1
        (cat(n))
        (synonyms "")
        (anno(def "agent of driving")(comments
"")(ex ""))
        (syn-struc((root($var0))(cat(n))))
        (sem-struc(human(agent-of(sem(drive)))))
)
```

The synonym list <human homo sapiens man> returned automatically for "human" by the name assignment procedure will be discarded because the (agent-of(sem(drive)) expression does not exhaustively encapsulate the meaning of the synonym list members. Thus, the acquirer will add the sense name manually:

```
(driver-n1
        (cat(n))
        (synonyms "")
        (anno(def "agent of driving")(comments
"")(ex "the driver stopped the car"))
        (syn-struc((root($var0))(cat(n))))
        (sem-struc(human(agent-
of(default(drive)))))
)
```

The synonym check procedure will search for a match of "driver" and return the list <automobilist, cabbie, chauffeur, motorist, operator>. Upon the acquirer's approval, six new noun senses will be generated by propagation:
automobilist-n1, cabbie-n1, chauffeur-n1, morotist-n1, operator-n1

Instrumental nouns

Since the variable ^$var2 on the INSTRUMENT is unrestricted, ontological fillers of the INSTRUMENT case role of the concept DRIVE are checked, and concepts CAR BUS TRACTOR and TRUCK are detected. The inverse of (drive(instrument(sem(car bus tractor truck military-tank)))) is derived, and each of the fillers and their children are checked for the expression (instrument-of(sem(drive))) in their definition. In the ontology where CAR, BUS, TRACTOR and TRUCK are subsumed under the same subclass LAND-DRIVEN-VEHICLE, the expression (instrument-of(default(drive))) justifiably contrasts it with siblings like AIR-VEHICLE, whose default INSTRUMENT-OF filler is FLY.

After (instrument-of(default(drive))) is found in the definition of CAR, BUS, TRACTOR and TRUCK, the enhancer will automatically create a sense template for each concept, removing duplicate ontological properties from their sem-strucs, e.g.:

```
(-n1
        (cat(n))
        (synonyms "")
        (anno(def "instrument of
driving")(comments "")(ex ""))
        (syn-struc((root($var0))(cat(n))))
        (sem-struc(car))
)
```

The name assignment procedure will establish the lexicalization for "car", "bus", "tractor" and "truck" by finding respective matches in the synonym list repository and automatically assigning names "car-n1", "bus-n1", "tractor-n1", and "truck-n1" to sense templates with respective sem-strucs.

After the acquirer's confirmation, the synonym check procedure will return appropriate lists for each match, e.g.
"car-n1": <auto, automobile, clunker, convertible, coupe, gas guzzler, hardtop, hatchback, jalopy, jeep, limousine, motor, motorcar, pickup, roadster, sedan, station wagon, subcompact, touring car>;
"bus-n1": <coach, autobus, omnibus, passenger vehicle, public bus>
"tractor-n1": <caterpillar>
"truck-n1": <lorry, eighteen-wheeler, semi, pickup>

Noun senses (nineteen for "car-n", six for "bus-n", two for "tractor-n" five for "truck-n" and their synonyms) are then created by rapid propagation. The sense of "car-n1", complete with synonyms, is listed below:

```
(car-n1
        (cat(n))
        (anno(def  "a land driven vehicle")
        (synonyms " auto-n1, automobile-n1,
        clunker-n1, convertible-n1, coupe-n1, gas-
        guzzler-n1, hardtop-n1, hatchback-n1,
        jalopy-n1, jeep-n1, limousine-n1, motor-n1,
        motorcar-n1, pickup-n1, roadster-n1, sedan-
        n1, station-wagon-n1, subcompact-n1,
        touring-car-n1")
         (comments "")(ex "Bill sold his old car"))
        (syn-struc((root($var0))(cat(n))))
        (sem-struc(car))
)
```

Deverbal noun senses

For "drive-v1", a deverbal noun "drive", as in "he wrote a review of the test drive of the new model by an expert", is available. First, derivation rules apply to the syn-struc of "drive-v1", where the syntactic subject is re-written as an optional prepositional adjunct with "by", and the direct object is re-written as an optional prepositional adjunct with "of". The optionality of the prepositional adjuncts is dictated by cases like "he took his first drive to the countryside", where both adjuncts are omitted. The sem-struc of "drive-v1"remains unchanged. The sense name is assigned automatically and approved by the acquirer, and a synonym list is retrieved: <ride, commute>. The complete sense is then derived:

```
(drive-n1
        (cat(n))
        (anno(def "an act of driving a land
vehicle")(ex "he took a drive to the countryside. he
reviewed the test drive of the new model by the
expert")(comments ""))
        (synonyms "ride-n1, commute-n1")
        (syn-struc((root($var0))(cat(n))
        (pp-adjunct((opt(+))(root(by))(cat(prep))
        (obj((root($var1))(cat(np))))))
        (pp-adjunct((opt(+))(root(to))(cat(prep))
        (obj((root($var2))(cat(np))))))))
        (sem-struc(drive(agent(value(^$var1)))
        (instrument(value(^$var2)))))
)
```

Adjectival senses

The sem-struc of the adjectival sense will be generated from the INSTRUMENT case role fillers of "drive-v1" by specifying the variable for the INSTRUMENT property inverse and assigning a high probabilistic (EPISTEMIC) modality value to the event DRIVE. Because CAR, BUS, TRACTOR and TRUCK are already specified as INSTRUMENT fillers of DRIVE in the ontology, they will be omitted from the variable restriction in the adjective sem-struc for reasons of redundancy.

Name assignment and synonym check procedures have limited application for adjective derivation for reasons described in 2.2(2); so the name "drivable", is assigned manually. The complete sense is listed below:

```
(drivable-adj
        (cat(adj))
        (anno(def "capable of being driven: of
cars")(comments "")(ex "Bill sold a completely
drivable car. This old car is still drivable"))
        (synonyms "")
        (syn-struc((root($var0))(cat(adj))
                (syn-np((root($var1))(cat(np))))))
        (syn-struc1((syn-np((root($var1))(cat(np))))
                (vp((root(be))(cat(v))))(adj-
        comp((root($var0))(cat(adj))))))
        (sem-struc(^$var1(restricted-to(car bus
tractor truck))(instrument-of(default(drive(epistemic
                        (value(0.8))))))))
)
```

4 References

[1] C. Hempelmann, J. Taylor; and V. Raskin, "Application-guided ontology engineering" In Proc. ICAI'10, Las Vegas, July 2010.

[2] J. Katz, and J. Fodor. "The Structure of a Semantic Theory". In *Language*, vol. 39, No. 2, pp. 170-210, April-June 1963

[3] S. Nirenburg and V. Raskin, Ontological Semantics. Cambridge, MA: MIT Press, 2004.

[4] V. Ooi. Computer Corpus Lexicography. Edunburg University Press, 1998.

[5] J. Preiss, T. Briscoe, and A. Korhonen. "A System for large scale acquisition of verbal, nominal, and adjectival subcategorization frames from corpora". In. Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics. Prague, Czech Republic, 2007.

[6] M. Petrenko, "Lexicon Management in Ontological Semantics". In Papers from the Annual international Conference "Dialogue 2010, Moscow, Russia 2010, in press.

[7] V. Raskin, J. M. Taylor, and C. F. Hempelman, "Guessing and knowing: Two approaches to semantics in natural language processing,". In Papers from the Annu. Int. Conf. "Dialogue", Moscow, 2010, vol. 9, no. 16, pp. 642-650.

[8] J. Taylor, C.F. Hempelmann, and V. Raskin, "On an automatic acquisition toolbox for ontologies and lexicons" In Proc. ICAI'10, Las Vegas, USA, July 2010.

Syntactic Transformations Modelling for Hybrid Machine Translation

E. Kozerenko
Institute for Informatics Problems of the Russian Academy of Sciences, Moscow, Russia

Abstract *The paper deals with the problems of cross-language syntactic transformations modelling for design and development of transfer-based machine translation systems. The solutions are proposed on the basis of the hybrid grammar comprising linguistic rules and statistical information about the language structures preferred in particular languages.*
Keywords: syntax, transformations, functional semantics, machine translation, cognitive transfer

1 Introduction

This work is dedicated to the vital problem of the realization of language transformations, mainly syntactical, which should be performed in the course of translation from one language into another. These transformations are implemented in the process of designing semantic-syntactic representations in the transfer-based machine translation systems and in intelligent systems for knowledge extraction from natural language texts. At present in the world practice the systems of statistical machine translation (SMT) [1-5]. prevail, however, further improvement of the methods of recognition and transfer of language structures is possible with the use in the language processors of such expert linguistic knowledge which give basis for machine learning on parallel texts. The creation of systemic representation of differences in the semantic-syntactic structures of the source and target languages is the most pressing issue.

The application of statistical models has considerably advanced the area of machine translation since the last decade of the previous century, however now new ideas and methods appear aimed at creating systems that efficiently combine symbolic and statistical approaches comprising different models.

Both the paradigms move towards each other: the procedures of analysis and translation are enhanced by the statistical data which are taken into consideration by the "translation engine" for disambiguation of language structures. The stochastic approach to natural language processing originates from the projects in speech and characters recognition and spellcheckers. The main method for solving numerous problems, including the part of speech establishment and tagging, is the Bayesian approach. The architecture of stochastic systems is based on the dynamic programming algorithm.

Machine learning is rooted in the stochastic research paradigm. The training algorithms can be of the two types: supervised and unsupervised. An unsupervised algorithm should infer a model capable for generalization of the new data, and this inference should be based on the data alone. A supervised algorithm is trained on a set of correct responses to the data from the training set so that the inferred model provides more accurate decisions. The object of machine learning is the automatic inference of the model for some subject area basing on the data from this area. Thus a system learning, for example, syntactic rules should be supplied with a basic set of phrase structure rules. The widely used methods lately have been the *N*-grams which capture many intricacies of syntactic and semantic structures [3-7], *N*-grams of variable length in particular, [6], introduction of semantic information into *N*-grams. In [4] a detailed description is given of the approach to creating a statistical machine translation based on N-grams of bilingual units called "tuples" and the four special attribute functions.

The statistical models are built on the data obtained from the parallel corpora in different languages. Usually the texts are compared within language pairs. The text in the language from which the translation should be done is called the *source text*, and the text which is its translation is called the *target text*. Correspondently the languages are also called the source language and the target language (i.e. the language of translation).

The main method of extracting the data about the matches between the source and target languages and texts is the alignment of parallel texts. The result of this procedure is also called *alignment* and it is designated by *A*. The probability characteristics of alignments are employed in the algorithms of statistical machine translation. Hence, the *alignment* and the probability distribution are the key notions in these models description.

The models under consideration are illustrated basing on the bilingual model for the Russian and English language pair. However, the similar methods are applicable for the alignments and translations of the Russian texts into the French and German languages, as well as other European languages.

2 Statistical and rule-based models of machine translation

The contemporary period of research and developments in the area of machine translation and knowledge extraction systems is characterized by the intensive process of the "hybridization" of approaches and models. The significant computational resources of the present-day systems make it possible to accumulate and to use the previously translated text fragments, to ensure machine translation based on precedents ("example-based

machine translation")[8] and to support effectively the component of "translation memory"[9]. Creators of the systems based on the rules introduce different stochastic models into the rules which make it possible to reflect dynamics and variety of language forms and values, and the supporters of statistical methods of constructing the linguistic models more frequently employ the approaches based on linguistic knowledge, considering this as the means "of intellectualization" of systems[10,11].

The statistical models of transfer are built on the basis of the data obtained from parallel texts corpora in different languages. Usually the comparison of texts is performed for language pairs. The basic method of obtaining the data about the correspondences between the source and target texts and the languages is the procedure of *alignment* of texts. The probability characteristics of *alignment* are used for creating the translation algorithms in the statistical models of machine translation. Thus, *alignment* and *probability distributions* are the key concepts for the description of these models. Statistical approaches to the alignment of parallel texts are aimed at finding the most probable version of the alignment for the given parallel texts *S* and *T*:

$$\arg\max_A P(A \mid S,T) = \arg\max_A P(A,S,T) \qquad (1)$$

In order to estimate the values of the probabilities which are indicated in this expression, the methods presenting the parallel texts in the form of a sequence of the aligned chains of sentences $(B_{1,\ldots,}B_K)$ are most frequently used, in this case it is assumed that the probability of one chain does not depend on the probabilities of other chains, but it depends only on the sentences in this chain [12,13], then:

$$P\ (\ A\ ,\ S\ ,\ T\ )\ \approx\ \prod_{k\ =\ 1}^{K}\ P\ (\ B_{\ k}\ ) \qquad (2)$$

This method simply considers the length of a sentence in the source language and in the language of translation, measured in symbols. It is assumed that the longer sentence of one language corresponds to the longer sentence of another language. This approach gives completely steady results for the similar languages and the literal translation. The more sensitive mechanisms of comparison are ensured by the methods of lexical alignment [14-16]. The model of transfer on the phrases, or the model of transfer on the basis of the templates of alignment [12,17], and other similar models very strongly advanced the development of the technology of machine translation due to the extension of the base units of translation from words to phrases, i.e., to the substrings of arbitrary size. However, very seldom the phrases of this model of statistical machine translation (SMT) are the phrases in the sense of any existing syntactic theory or formal grammar.

3 The alignment on the basis of the Cognitive Transfer Grammar

In the cognitive transfer grammar (CTG), formulated in [17,18], the functional values of language structures are determined by the categorial values of head vertices. Probability characteristics are introduced into the rules of derivation in the form of the weights assigned to the parse trees. In CTG the elementary structures are *transfemes*. *Transfeme* is the unit of cognitive transfer, which establishes the functional-cognitive correspondence between the structures of the source language L_s and the structures of the target language L_T. For the alignment of parallel texts transfemes are given as the rewriting rules in which a nonterminal symbol is in the left side, and right sides contains the aligned pairs of chains of terminal and nonterminal symbols belonging to the source and target languages:

$$T \rightarrow \langle \rho, \alpha, \square \rangle, \qquad (3)$$

where T is a nonterminal symbol (transfeme), ρ and α are the chains of the terminal and nonterminal symbols which belong to Russian and English, $\square$ is the symbol of correspondence between the nonterminal symbols, occuring in ρ and with the nonterminal symbols occurring in α. During the alignment of parallel texts on the basis of cognitive transfer grammar the process of derivation begins from the pair of the connected initial symbols S_ρ and S_α, further at each step the connected nonterminal symbols in pairs are copied with the use of two components of uniform rule. For the automatic extraction of rules from the parallel texts on the basis of cognitive transfer grammar it is necessary to preliminarily align texts by the sentences and words. The extracted rules rest on the word-by-word alignments in such a way that at first the initial phrase pairs are identified with the use of the same criterion as in the majority of the statistical models of transfer, based on the phrase approach, namely: there must be at least one word inside the phrase in one language, aligned with a certain word inside the phrase in the other language. The existing methods basically employ either sentence alignment or word alignment some experiments are made with phrase alignment and recently a mixed sentence-word approach has been developed to explore the paraphrases in the aligned parallel corpora. These attempts to consider linguistic information mark a step forward to acknowledging the intricate character of natural language if compared with other types of data. The mixed approach employs both sentence and word alignments. However, all these methods deal with the structural elements without considering the semantic aspects of the aligned language units.

The phrase-based translation model, or the alignment template model [12] and other similar approaches have greatly advanced [14] the development of machine translation technology due to the extension of the basic translation units from words to phrases, i.e. the substrings of arbitrary size. However, the phrases of this statistical machine translation model are not the phrases in the meaning of any existing syntax theory or grammar formalism, thus, for example, a phrase can be like "alignments the", etc. A real challenge is the cross-level (e.g. morphology-to-syntax) matching of language structures in parallel texts [21]. New research and development results demonstrate the growing awareness of the demand for enhancing linguistic motivation in statistical translation models and machine learning techniques [22].

4 Intertext development: establishment of of semantic matches

The above stated methods are being employed for design and development of a linguistic knowledge base Intertext. It is a linguistic resource with semantic grouping of phrase structure patterns provided with the links to synonymous structures at all language levels for the languages included into the linguistic base.

Our focus on configurations provides high portability to the language processing software designed under these principles: we can operate with a lexicon which has only standard linguistic information including morphological characteristics, part of speech information and the indication of transitivity for verbs.

The Intertext linguistic knowledge base comprises the following components:

- parallel texts database: the texts are segmented into the functionally relevant structures that are semantically aligned;
- a bilingual Treebank (under development at present);
- structural parse editor (under development at present) which displays the parse and transfer schemes for indicated text segments;
- the inventory of structural configurations arranged on Cognitive Semantic principle.

5 Establishment of cross-language matches and inter-structural synonymy

Translation activity involves the search for equivalence between structures of different languages. However, to establish whether the structures and units are equal or not, we need some general equivalent against which the language phenomena would be matched. Our approach based on the principle "from the meaning to the form" focusing on Functional Syntax and Cognitive Semantics would yield the necessary basis for equivalence search.

5.1 Types of matches

The following types of structural semantic matches have been observed:

word → word, phrase structure → phrase structure, word → phrase structure, morpheme → word, morpheme →phrase structure.

Syntactically languages are most different in the basic word order of verbs, subjects, and objects in declarative clauses. English is an SVO language, while Russian has a comparatively flexible word order. The syntactic distinction is connected with a semantic distinction in the way languages map underlying cognitive structures onto language patterns, which should be envisaged in MT implementations

The basis of Cognitive Transfer Grammar (CTG) is composed of the proto-typical structures of the languages (in the initial model Russian and English) being investigated, their most probable positions in a sentence, statistical data about the distributive characteristics of structures (the information about the contextual conditions of the use of the investigated objects, i.e. the information about the structural contexts), the schemes of the complete parse of sentences.

The creation and development of the CTG assumes:

- the semantic approach to the analysis of language meaning and language form (forms);
- the construction of formal grammar presentations taking into account the structures of components and mechanisms of linearization, and also the relations of dependence between the units of a syntactic tree (the approach, which has the features of similarity to HPSG: the inheritance of the features via the head elements of phrase structures);
- the inclusion of the probability characteristics of language objects;
- the creation of Cognitive Transfer Spaces (CTS), represented in the form of expert linguistic rules, which can be extended by means of the establishment of synonymous language structures of parallel texts in different languages. The notion of Cognitive Transfer Spaces is the elaboration of the Functional Transfer Fields idea [17] for the multivariant translations of language structures.

In contrast to the approaches on the basis of "translation memory" that provide the increase of a machine translation system language competence by accumulating the previously translated text fragments and mainly based on regular expressions, Cognitive Transfer Grammar is intended for the realization of the mechanism of structural memory, which simulates language competence of an adult learner ("Adult Learning Memory"). Thus, structural memory comprises the following components:

1) The initial basic collection of grammar rules represented in the formalized form (CTG);

2) The mechanisms of expansion and refinement of the system of rules, implemented by means of the methods of machine learning on parallel texts.

Our studies are based on the concepts of the functional approach, which we have used for the multilingual situation. With the development of the linguistic processor, which ensures English - Russian and Russian - English transfer, we introduced the concept of functional transfer fields (FTF) [17,18] that served the basis for the segmentation of language structures for the solution of machine translation problems. The basic idea of FTF consists in the adoption of the hypothesis about the fact that at the basis of grammatical structures there lie the cognitive structures (mental frames); a functional transfer field reflects the interaction of elements from different language levels.

The basic design unit of the spaces of cognitive transfer is a *transfeme*.

Definition. *Transfeme* is a unit of cognitive transfer the, i.e. a semantic element embodied in a translatable semantically relevant language segment taken in the unity of its categorial and functional characteristics, that establishes the semantic correspondence between the language structures, which belong to different language levels and systems. The types of transfemes are determined by the rank of transfemes.

We distinguish the following ranks of transfemes:

• rank 1: lexemes as structural signs, i.e., a word, considered as a categorical - functional unit without taking into account the specific lexical value of this word;

• rank 2: a word combination, i.e., the syntactic structure, which consists of two and more syntactically connected words, but never a complete sentence (clause);

• rank 3: a clausal unit, i.e., dependent (subordinate) clause;

• rank 4: a sentence (either a simple sentence or the main clause of a complex sentence);

• rank 5: a scattered structure, i.e., a word group, which is characterized by a syntactic and semantic unity, but is discontinuous, i.e., between the members of the group there appear other language objects, which are not the members of this group;

• rank 0: the morphological units, which are not independent words, but which form a part of a lexeme of a source language, and in the language of transfer can be expressed by a clause and the units of other ranks, for example: the suffixes – *ible, - able* which are synonymous to the construction *"which can be"*, e.g. *extensible – which can be extended.*

5.2 Cross-level focus

Our studies also focus on particular situations when the semantic match goes across language levels. The segmentation of phrase patterns used for the input language parse was carried out with the consideration of semantics to be reproduced via the target language means. Both the most important universals such as enumeration, comparison, modality patterns, etc., and less general structures were singled out and assigned corresponding target language equivalents.

Consider an example of a phrase structure conveying the modal meaning of obligation: "*…the task to be carried out…*". In other words, the meaning of this phrase can be rendered as "*…the task that should be carried out…*". The Infinitive phrase in the English language gives the regular way of expressive means compression without the loss of semantic value. A literary translation in Russian requires the second way of presenting the same idea of obligation. However in this specific case a "reduced" translation variant is also possible which consists in the introduction of the subordinate conjunction "*chtoby*" – "*so that*", between the noun and the modifying Infinitive. The parse rule would look like: *NP(to)* → *NP VPto; a*nd the generation rule would be presented as: *NP(to)* → *NP Punct.{comma} Conj.(chtoby) VPto*.

Special attention is required for the problem of passive constructions transfer. As in the phrase "*was considered*". The rules for simultaneous translation (which in many cases is similar to the real time machine translation performance and can be a source of compromise decisions for phrase structure design) requires the transformation of the English Subject into the Direct Object (Russian, Accusative Case) standing in the first position in a sentence and the passive verbal form would produce an impersonal verbal form in Russian. Actually the process of transfer goes across the functional – categorial values of language units. A language structure which can be subjected to transfer has to be semantically complete from the point of view of its function. The cases of categorial shifts, in particular, when the technique of conversion is employed, require special treatment: the categorial shift of a syntax unit is determined by the functional role of this unit in a sentence (e.g. noun as a modifier → adjective).

Sometimes, a word may be translated by a word of another part-of-speech in the target language, a word combination, or even a clause, as the English *implementable* is best translated into Russian as *kotoryi vozmozhno realizovat* (*which can be implemented*). To overcome these differences the categorial and functional features of the two languages were considered, and the structures of the input were made conformed to the rules of the target language by applying contrastive linguistic knowledge for implementation of the transfer model. A suitable formalism is indispensable for an algorithmic presentation of the established language transfer rules, and the language of Cognitive Transfer Structures (CTS) was developed based on rational mechanisms for language structures generation and feature unification.

We apply the multivariant CTG constraints to our parse and transfer algoritm to choose the optimal variants for translations from English into Russian (and from Russian into English). Each phrase (transfeme) has a set of different CTG labels, and we need a way of choosing which label to use when applying the constraint. At present we

choose the best label for the phrase in a parse tree and the best transfer variant in the language of translation:

$$e = \arg\max_{e} \arg\max_{s \in CTG-labels(e,P)} p(e \mid r, s) \quad (4)$$

where *e* is an Enlish sentence, *r* is a Russian sentence, *P* is an English parse tree, *s* is a syntactic type of e belonging to the Cognitive Transfer Grammar.

Our linguistic simulation efforts are aimed at capturing the cross-level synonymy of language means and cross-linguistic semantic configurational matches for the English and Russian languages. The emphasis on the practical human translation experience gives the reliable foundation for statistical studies of parallel text corpora and automated rule extraction in further studies.

5.3 Polysemy of syntactic structures

By syntactic polysemy we mean the immediate realization of *more than one categorial meaning within the head* element of a language structure. The polysemous structures display variable manifestation of their categorial features depending on the functional role in the sentence. Consider such language phenomena as the Gerund, the Participle and the Infinitive. The Gerund comprises the features of both the Verb and the Noun, which affects the translation strategy when the appropriate means are to be chosen for representation of the English Gerund via the Russian language forms. The structures similar in category to the English Gerund are the Russian Verbal Nouns denoting "Activity", e.g. sing*ing* → pen*ie,* read*ing* → chten*ie*, and both the English Gerund, and the Russian Verbal Noun allow direct object arguments if derived from transitive verbs. However, the direct transfer of the Gerund into the Russian Verbal Noun is the least probable translation variant of the three possible transfer schemes:
The Gerund (Eng) → Clause with the Finite Verb form (Rus)
The Gerund (Eng) → Clause with the Infinitive
The Gerund (Eng) → Verbal Noun.
Consider the other most productive polysemous language structures which comprise more than one categorial meaning:
The Participle → Verb + Adjective
The Infinitive → Verb + Noun
Nominal Phrase (Nominal Modifier) →Noun + Adjective
Verbal Phrase (Verbal Modifier) → Verb + Adverb.

Thus we introduce the notion "polysemous syntactic structure" to determine the set of possible transfer schemes for a given language structure. When a polysemous structure is assigned specific categorial attributes realized in this structure, the possible and preferable transfer schemes become predictable for the given structure. The predominant categorial meaning of a polysemous syntactic structure is determined by the syntactic function realized at a given moment. Thus the transfer scheme for a "*stone wall*" construction will be as follows:

Noun1 + Noun2 [Eng.] →Adjective + Noun2 [Rus]

The weight for this transformation will be higher than for the transformation:
Noun1 + Noun2 [Eng] → Noun2 + Noun1 (Genitive) [Rus]
if the dictionary contains an Adjective as one of the possible translation equivalents for Noun1.

We find it important to differentiate between polysemous and ambiguous syntactic structures. A *polysemous* structure implies possible realizations of meanings which are compatible within one language structure and can be transferred to the structures of another language which are isofunctional to the source language structure. An *ambiguous* syntactic structure presupposes alternative ways of interpretation, the meanings being incompatible within one language structure, thus we deal with ambiguity when we try to discern some Finite and Nonfinite verbal forms:
Gerund / Present Participle;
Infinitive / Present Simple;
Past Participle / Past Simple.

Ambiguous structures can be misleading to the parsing procedures and subsequent machine translation, as for example, the "garden path" is a well-known language phenomenon which may give incorrect parse at the early stage of analysis, that could be corrected only at the final stage:
The cars passed by the vessel drowned.
The possible interpretations for the sentence can be
The cars which were passed via the vessel drowned (the correct variant).
The cars which passed the vessel drowned.

6. Rule set for training data: functional-cognitive approach

The establishment of structures equivalence on the basis of functional semantics proved to be useful for developing the syntactic parse and transfer rules module for the English – Russian machine translation. This rule module was implemented in the first release of the Cognitive Translator system [17]. Generally, major efforts connected with natural language modeling lay emphasis at lexical semantics presentations and less attention is paid to the semantics of structures and establishment of functional similarity of language patterns as a core problem in multilingual systems design.

The set of functional meanings together with their categorial embodiments serves the source of constraints for the unification mechanism in the formal presentation of our grammar. The formalism developed employs feature-based parse, and head-feature inheritance for phrase structures which are singled out on the basis of functional identity in the source and target languages. The transferability of phrase structures is conditioned by the choice of language units in the source and target languages belonging to the same functional transfer fields (FTF), notwithstanding the

difference or coincidence of their traditional categorial values. A set of basic FTF was singled out and language patterns employed for conveying the functional meanings of interest were examined:

- Primary Predication FTF (non-inverted) bearing the Tense – Aspect – Voice features; this field mainly includes all possible complexes of finite verbal forms and tensed verbal phrase structures.
- Secondary Predication FTF bearing the features of verbal modifiers for the Primary Predication FTF. Included here are the non-finite verbal forms and constructions, and subordinate clauses comprising the finite verbal forms. All these are united by the functional meanings they convey, e.g. qualification, circumstance, taxis (ordering of actions), etc.
- Nomination and Relativity FTF: language structures performing the nominative functions (including the sentential units) comprise this field.
- Modality and Mood FTF: language means expressing modality, subjunctivity and conditionality are included here. Here the transfer goes across the regular grammatical forms and lexical means (modal verbs and word combinations) including phrasal units.
- Connectivity FTF: included here are lexical – syntactic means employed for concatenation of similar syntactic groups and subordination of syntactic structures.
- Attributiveness FTF: adjectives and adjectival phrases in all possible forms and degrees comprise the semantic backbone of this field; included here are also other nominal modifiers, such as nominative language units and structures (*stone wall* constructions, prepositional genitives – *of* –phrases), and other dispersed language means which are isofunctional to the backbone units.
- Metrics and Parameters FTF: this field comprises language means for presenting entities in terms of parameters and values, measures, numerical information.
- Partition FTF: included in this field are language units and phrase structures conveying partition and quantification (e.g. *some of, part of*, *each of*, etc.).
- Orientation FTF: this field comprises language means for rendering the meaning of space orientation (both static, and dynamic).
- Determination FTF: a very specific field which comprises the units and structures that perform the function of determiner (e.g. the Article, which is a good example for grammar – lexical transfer from English into Russian, since in Russian there exist no such grammatical category; demonstrative pronouns, etc.).
- Existentiality FTF: language means based on *be*-group constructions and synonymous structures (e.g. sentential units with existential *there* and *it* as a subject: *there is*…; *there exists*…; etc.).
- Negation FTF: lexical – syntactic structures conveying negation (e.g. *nowhere to be seen*, etc.).
- Reflexivity FTF: this field is of specific character since the transfer of reflexivity meaning goes across lexical - syntactic – morphological levels.
- Emphasis – Interrogation FTF: language means comprising this field are grouped together since they employ grammar inversion in English.
- Dispersion FTF: individual language structures specific for a given language are included here; these are presented as phrasal templates which include constant and variable elements. We single out 3 major types of the Dispersion FTF.

Interpretation techniques employ the segmentation of structures carried out on the basis of the functional transfer principle. The principal criterion for including a language structure into a field is the possibility to convey the same functional meaning by another structure of the field. A constraint-based formalism which is called the Multivariant Cognitive Transfer Grammar has been developed and. It comprises about 350 transferable phrase structures together with the multiple transfer rules combined within the same pattern - Cognitive Transfer Structures (CTS), which are statements of the declarative syntactical processor module and encode both linear precedence and dependency relations.

However, we see our main objective not in the creation of an abstract semantic meta language, but in a careful research of all possible kinds of configurations of language patterns used by natural languages for the expression of functional-cognitive meanings.

7 Conclusion

The key idea of the linguistic framework presented in the paper is cognitive cross-linguistic study of what can be called *configurational* semantics, i.e. the systemic study of the language mechanisms of patterns production, and what meanings are conveyed by the established types of configurations. We explore the sets of meanings fixed in grammar systems of the languages under study. Our studies are focused on the types of meanings outside the scope of lexical semantics, and we consider the lexical semantics when the meanings which we denote as configurational, have expression at the lexical level. The importance of this aspect is connected with the fact that natural languages are selective as to the specific structures they employ to represent the referential situation. However, it is always possible to establish configurations which perform the same function across different languages (i.e. isofunctional structures). The parse aimed at transfer procedures requires a semantic grammar and cannot be efficiently implemented through a combination of monolingual grammars. The absence of the complete agreement between the English and Russian language constructions in the scientific and technical texts can be revealed during the study of the comparative frequency of different parts of speech employment, which is important for constructing transfer systems using machine learning. The need for the

simulation of language transformations for the systems of machine transfer and extraction of knowledge from the texts is caused by the fact that, until now these phenomena little are investigated from the point of view of the possibilities of their computer realizations and it is, correspondingly, insufficiently taken into account in the working systems. With the formation of the procedures of disambiguation the statistical data is used obtained during the study of the parallel texts of scientific and patent documents in our experimental corpus. Thus, for instance, with the ranking of the preferences of the transfer versions we use the statistics of correspondence of finite and non-finite verb forms and transformations on them, the priorities of the use of active or passive constructions, nominalizations and verbalizations in Russian, English and French discourse. Our further studies are connected with the expansion of the number of types of transformations in the linguistic representations for the multilingual situation, the creation of the engineering linguistic medium of research and developments in the area of machine transfer and extraction of knowledge from the texts in different languages.

References

[1] Brown P. F., Cocke J., Della Pietra S. A., Della Pietra V. J., Jelinek F., Lafferty J. D., Mercer R. L., Roossin P. S. A statistical approach to machine translation // Computational Linguistics, 1990. Vol. 16. P. 79–85.

[2] Brown P. F., Della Pietra S. A., Della Pietra V. J., Mercer R. L. The mathematics of statistical machine translation: Parameter estimation // Computational Linguistics, 1993. Vol. 19. No. 2. P. 263–311.

[3] Rosenfeld R. A maximum entropy approach to adaptive statistical language modeling // Computer Speech and Language, 1996. Vol. 10. P. 187–228.

[4] Marino J. B., Banchs R. E., Crego J. M., de Gispert A., Lambert P., Fonollosa J. A. R., Costa-Jussa M. R. N-gram-based Machine Translation // Computational Linguistics, 2006. Vol. 32. No. 4. P. 527–549.

[5] Knight, K. Automating knowledge acquisition for machine translation. AI Magazine 18, pp.81-96, 1997.

[6] Niesler T. R., Woodland P. C. Modelling word-pair relations in a category-based language model // IEEE ICASSP-99, IEEE, 1999. P. 795–798.

[7] Ney H., Essen U., Kneser R. On structuring probabilistic dependencies in stochastic language modeling // Computer Speech and Language, 1994. Vol. 8. P. 1–38.

[8] Brown, R.D. Example-based machine translation in the Pangloss system // COLING-96, Copenhagen, pp. 169-174, 1996.

[9] Compendium of Translation Software. John Hutchins, editor, Seventh edition, August, 2003. Lagoudaki, E. Translation Memories Survey 2006: User's Perceptions Around TM Usage // Proceedings of the Translating and the Computer 28 Conference, 16-17 November 2006, London. Aslib/IMI, London, 2006, pp. 1-29.

[10] Wang, Ye-Yi, and Alex Waibel. Modelling with structures in statistical machine translation // ACL 36/COLING 17, pp. 1357-1363, 1998.

[11] Plana, E. SIMILIS Second-Generation Translation Memory Softwar // Proceedings of the Translating and the Computer 28 Conference, 25-24 November 2005, London. Aslib/IMI, London, 2005.

[12] Och F.J., Ney H. The alignment template approach to statistical machine translation // Computational Linguistics, 2004. Vol. 30. P. 417–449.

[13] Gale, William A. and Kenneth W. Church. A program for aligning sentences in bilingual corpora // Computational Linguistics. 1993. Vol. 19. P. 75-102.

[14] Koehn P. and Hoang H. Factored translation models // Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL). 2007. P. 868–876.

[15] Chen, Stanley, F. Aligning sentences in bilingual corpora using lexical information // Proceedings of the 31st Annual Conference of the Association for Computational Linguistics. 1993. P. 9-16.

[16] Melamed, I. Dan. Models of translational equivalence among words // Computational Linguistics. 2000. Vol. 26, No 2. P. 221–249.

[17] Kozerenko, E.B. Cognitive Approach to Language Structure Segmentation for Machine Translation Algorithms // Proceedings of the International Conference on Machine Learning, Models, Technologies and Applications, June, 23-26, 2003, Las Vegas, USA.// CSREA Press. 2003. P. 49-55.

[18] Kozerenko, Elena. Features and Categories Design for the English-Russian Transfer Model // Advances in Natural Language Processing and Applications Research in Computing Science. 2008. Vol. 33. P. 123-138.

[19] Callison-Burch C. Syntactic Constraints on Paraphrases Extracted from Parallel Corpora // Proceedings of EMNLP-2008. 2008.

[20] Callison-Burch C., Koehn P., Monz C., Schroeder J. Findings of the 2009 Workshop on Statistical Machine Translation // Proceedings of Workshop on Statistical Machine Translation (WMT09), 2009.

[21] Yeniterzi R. and Oflazer K. Syntax-to-Morphology Mapping in Factored Phrase-Based Statistical Machine Translation from English to Turkish // Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 454–464, Uppsala, Sweden, 11-16 July 2010.

[22] Machine Translation and Morphologically-rich Languages Research Workshop of the Israel Science Foundation, University of Haifa, Israel, 23-27 January, 2011, website: http://cl.haifa.ac.il/MT/program.shtml

Semantic Approach to Explicit and Implicit Knowledge Extraction

I.P. Kuznetsov, E.B. Kozerenko
Institute for Informatics Problems of the Russian Academy of Sciences, Moscow, Russia

Abstract *A semantic approach to constructing the linguistic processor which extracts the explicit and implicit objects and their links from natural language texts is considered. It is intended for the areas where the automatic formalization of the flows of texts in natural language is required: summaries of incidents, information in the criminal cases, archive materials, resume, mass media issues, and other texts. The use of this processor for text formalization in different subject areas is considered.*
Keywords: semantics, natural language, linguistic processor, knowledge extraction, named entities, implicit information

1 Introduction

One of the primary tasks in the area of cognitive technologies is the automatic extraction of knowledge from natural language (NL) texts. It is a complex problem connected with developing linguistic processors which perform automatic formalization of texts, i.e. mapping the texts into formal models or knowledge structures. It should be noted that a lot of relevant information in NL texts is presented in a concealed form. This information is called implicit. An example is the task of assigning certain features to persons on the basis of actions performed by them. In the subject area of "Criminology" it is assigning such features as "victim", "suspect", etc. to persons. The methods of implicit information extraction should be considered in the context of the knowledge extraction task, and it is conditioned by the specific features of the linguistic processor. The given paper describes these methods within the framework of the object-oriented linguistic processor developed at the Institute for Informatics Problems of the Russian Academy of Sciences (IIPRAS).

2 The linguistic processor for object extraction

The research direction connected with unstructured NL texts processing has been developing for 20 years at the IIPRAS for particular application areas and specific user tasks [1,2]. One should consider that the large category of users have the specific official responsibilities, and respectively, constant interests. Completely concrete information is necessary for them. For example, a criminal inspector seeks to extract information on important figurants, their places of residence, telephones, criminal events, dates and other such facts [3]; a personnel manager is interested in the organizations, when and where a person worked and in what position [4]. Other people try to fish out from the media the information about the countries, important persons, catastrophes, places of interest and historical monuments [5]. We call this concrete information interesting for a user *information objects*. Objects are distinguished by their types.

Let us note that the connections between the objects, which interest users, can have the high degree of variety. For example, not only the connection of the persons with their information from their questionnaires or the objects can present interest, but also the actions or the events, in which these persons participate. Such events are attached to the time and the place. Moreover, some events can be a component part of others. They can be connected with cause-effect and temporary relations. For the number of problems similar connections play an important role. They also must be revealed and processed. Therefore one should consider that events are also information objects, interconnected and connected with other information objects. Complex structures appear. For their representation within the framework of the projects of IIPRAS (Russian Academy of Science) the language of the extended semantic networks (ESN) has been developed, while for the processing the production rules the language DEKL [6] has been implemented.

The ESN networks are represented in the form of special graphs [6]. In the formal record they are the extension of the predicate logic language. ESN consist of elementary fragments, each of which has its unique code (see Section 3), which can stand at the argument places of other fragments, and provide great possibilities for the representation of knowledge structures. The language DEKL is designed for the transformation of such structures. The problem of extracting knowledge from natural language texts is considered from the point of view of developing information objects and connections with the construction of the knowledge structures on the basis of which the solution of user problems is achieved. For this within the framework of the IIPRAS projects the *object-oriented* linguistic processor (LP) converting the natural language (NL) texts to the knowledge structures is developed and constantly updated. The processor LP achieves a deep NL text analysis with bringing of synonymous groups to one form, development of objects and their properties, identification of objects, elimination of ambiguities, development and unification of various forms, which present events or actions (including forms with the verbal noun, participial and verbal-adverbial constructions), which are connected with the time and the place [2-6]. As a result the structures of knowledge in the ESN formalism are created.

The linguistic processor (LP) is realized by means of the language DEKL and is controlled by the linguistic knowledge (LK) in the form of object dictionaries, means

of parametric tuning, and also the rules of extracting objects and connections [2,4,5,6]. With the aid of LK the tuning of LP to the appropriate categories of users and text corpora is accomplished. Concrete realization appears as a result. Thus, the paper deals with the means of constructing a class of processors with powerful mechanisms for their tuning and updating. Further development of such processors (LP) is connected with the development of implicit information [7], which we will consider in a narrow plan, i.e. as the addition of the structures of knowledge by the new information, which is absent or assigned implicitly. In this article the procedure of this development is proposed, which consists in the use of LP for mapping NL texts onto the structures of knowledge (ESN) and the use of the means of logical-analytical processing (productions of the language DEKL) for the creation of new information.

Advantages and deficiencies of the proposed procedure will be examined on specific objectives from the area of "criminology", that is the role functions establishment for the persons (participants) on the basis of the acts performed by them or due to the participation in some specific events. We consider the problem of assignment of properties to the persons (basing on their participation in the acts of different kinds) - "the suffered", "the suspect" and others, if an explicit description of such properties is absent from the text. For example, if it is said in the text "suffered Ivanov I.I.", then another task appears, i.e. extraction of some property in the process of linguistic analysis and forming of the corresponding fragments in the knowledge structure. In this article the discussion will deal with LP, customized for the Russian language texts (NL), although the possibilities of LP are wider. There is a sufficient test of tuning LP to the English language texts [9,10].

2 The design method

The task of the role functions establishment for the information objects is a special case of the more general task, connected with the estimation of objects according to their descriptions in the NL texts, for example, with the estimation of the stability of enterprise (according to the information from the Internet), by featuring political figures (positive or negative depending on the statements in the press), by the estimation of the role functions quality of product (basing on the statements of users) and so forth. Quite frequently, it is not said directly whether something is bad, or good. As a rule, in NL texts the events are described, the situations, in which one or other information object participated. On the basis of them the estimation is done, which is often represented in the form of a new (generated) property of object.

For the solution of this problem different methods are used [11-15]. The most common one is the method of the new properties of objects development by using the syntactical-semantic forms. For example:

<what-medicine> caused allergy in <who-human organism>…,

< what-medicine > has side effects …

<who-person> made scandal… and so forth.

The application of such forms to the NL texts consists in the search for "estimating" or "characterizing" words (of type "scandal") or for word combinations of the type "caused allergy" ("it can cause allergy"), it "has side effects" ("side-line actions"), "to make scandal" ("to brawl")… And then the environment is analyzed, i.e., the words, which stand to the left and to the right, their semantic classes (objects are recognized by them) and case forms. Estimations of information objects as a result are given. By the first two forms the "quality of medicines" is estimated, while by the latter it is recognized that a man performed "hooligan actions" or that he is "suspected". It is known that in NL many versions are possible for expressing the same idea - with the aid of different syntactic constructions, verbal groups, forms and so forth. Therefore the number of estimating word combinations will be sufficiently large. Moreover, the application of such forms requires different forms of analysis - morphological (in order to reduce different word forms to one form), syntactic (the trees are built of the selection of sentences in order to isolate the connected components and to find place for the estimated words) and semantic (in order to extract the objects, which are evaluated). The use of syntactical-semantic forms is connected with certain difficulties caused by special NL features: by the presence in texts of participial, verbal-adverbial constructions, different explanations, facultative components (time, place, purpose), anaphoric references and other language structures. As a result, information objects are frequently disconnected from the estimated words. Hence - the significant losses, which influence the quality of estimation.

Example 1 *(the text is taken from the summaries of incidents of the City Office of Home Affairs, Moscow):*
… Gorelov Peter Sergeyevich, 01.03.76 yr/bir, liv: c. Moscow, st. Young Leninists, h.71-6-12, does not work, 01.02.1998 yr. at 4.30 in his house out of hooligan motives in the state of alcoholic intoxication made scandal and broke window glass in the apartment of Litvinova Galina Ivanovna, 20.07.1961 yr/bir,…

In this example the estimating (characteristic) words are "made scandal" and "broke the window glass", they are located at a significant distance from the estimated person - "Gorelov Peter Sergeyevich". This limits the possibilities of applying the forms. It is required that the initial extraction of components, which must not be considered in the forms: the years of birth, addresses, specific properties ("he does not work", "in the state of alcoholic intoxication"), time, place and others, which requires sufficiently deep text analysis with the extraction of objects, their properties and attributes. In connection with the aforesaid, another more promising method is represented - when evaluation is accomplished at the level of knowledge structures. For their construction the objective-oriented LP is used producing the structures of knowledge in which the objects are directly connected with the events and the actions and excluding the above mentioned losses. For the development of implicit

information (role functions of objects) the rules of the DEKL language are used which analyze the structures of knowledge (ESN) and form new properties of objects. In this case the structure of knowledge does not change, but it is only supplemented by new (useful) fragments.

3 Presentations of the meaningful portraits of documents

Within the framework of the proposed procedure the development of the role functions of objects (implicit information) is achieved at the level of the structures of knowledge, called the meaningful portraits of documents (SS-documents). Let us examine how such structures appear in the ESN formalism [2,3,6].

Example 2 (translation of the Russian text given below). *Text N22 is taken from the summaries of incidents of the City Office of Home Affairs, Moscow*:

01.02.98 yr. 16-30 to the Home Office applied citizen Mitrofanov Victor Mikhaylovich, 1955 yr. bir., liv.: Bohr Highway 38-211, n/w. he stated that 01.02.98 yr. at 10-00 in house 3 at St. Fedosino the unknowns being found in the drunk state made scandal, they expressed themselves by unquotable swearing, they set dog. As a result of what Mitrofanov applied to trauma care center, where the diagnosis was set: the bite of foot.

The objective-oriented LP performs the deep analysis of the text and automatically builds its meaningful portrait (SS- document, transliterated):

DOC_(22, "1-02-98", "SUMMARY; " /0+) 0 (RUS)
OVD_(OVD/1+)
FIO(MITROFANOV], VICTOR, MIKHAYLOVICH, 1955/2+) UNEMPLOYED (2-/3+) 3- (22, PROP)
ADR_(Borovskiy, Sh., 38,211/4+)
PROZH. (it is 2nd, 4)
ADR_(UL, FEDOSINO, HOUSE, 3/5+)
FIO (" " , " " , " " , NESKOLKO/6+)
UNKNOWN (6)
DRUNK (6-/7+) 7 (2, PROP_)
SCANDAL (6, PYANYY/8+)
IS EIGHTH (22, ACT_)
TO REPORT (IT IS 2ND, 8-/9+) 9 (22, ACT_)
DATA_(1998,02, ~01, " 10-00" /10+)
When (9, 10)
TO TURN (1, GR- N, 2-/11+) 11- (22, ACT_)
DATA_(1998,02, ~01, " 16-30" /12+)
When (11-, 12-)
EXPRESS (6, UNQUOTABLE, [BRAN]/13+) 13- (22, ACT_)
TO SET (6, [SOBAKA]/14+) 14 (0, ACT_)
TO TURN (IT IS 2ND, IN, [TRAVMPUNKT]/14+) 14 (0, ACT_)
TO PLACE (DIAGNOSIS, BITE, [NOGA]/16+) 16 (0, ACT_)
PREDL_(22,11-, 4, 3-, 9, 13-, 14-/17+) 17- (2,15,341)
PREDL_(22,15-, 16-/18+) 18- (6,342,448)

A meaningful portrait consists of the elementary fragments, arguments of which are words in the normal form (necessarily for the search and processing). Each elementary fragment has its unique code, which is written in the form of the number with the sign + and is separated by a slash line. For example, in the fragment OVD_(OVD/1+) the sign 1+ is its code (but 1 is the reference to it). Fragments DOK_(22, "1-02-98.TXT", "SUMMARY; " /0+) 0 (RUS) indicate that the meaningful portrait is built on the basis of the Russian-language text of document with number 22 of the file of 1-02-98.TXT", which was processed as the summary of the incidents (linguistic knowledge depend on this). The following fragments present police department OVD_(… /1+), person's surname, name and patronymic FIO (… /2+), person's specific property UNEMPLOYED (2-/3+), address ADR_ (… /4+) and so forth; the signs 2+, it is 2nd, 3+, 3-,… are the codes of the fragments, with the aid of which their connections and relations are assigned. For example, the fragment PROZH (live) (it is 2nd, 4) represents the relation that the person (represented as FIO with code 2+) lives at the address (fragment [ADR]_ with code 4+). Actions are represented in the form of fragments of the type SCANDAL (6, PYANYY/8+) it is 8 (22, ACT_), where it is represented that "person (FIO with code 6+), being drunk, made scandal". With the aid of it is the fragment 8_(22, ACT_) indicates that the first fragment is SCANDAL (…./8+) presents the action and relates to the document with the number 22. A similar role is played by the fragments of the type 3-(22, PROP_), by which the properties are noted. The codes of fragments also serve for the idea of time, scene of action and cases, when one action is included in the composition of another. For example, the fragment TO REPORT (it is 2nd, 8-/9+) represents that the person (code 2+) "reported" (code 9+) about the action (code 8+), i.e., about "made scandal". The following fragments DATA]_(… /10+) when (9, 10) represent the time (DATA_), which relates (when) to the action "to report". Special role is played by the fragments PREDL_(...), which correspond to the sentences. They are filled up with the words, which did not enter the information objects (in this example they are absent), or with the codes of objects themselves. To these fragments the indicators of their position in the text are added. For example, the fragment PREDL_(22,11-, 3-, 9, 13-, 14-/17+) 17- (2,15,341) represents the fact that the objects with codes 11- (corresponding to the action "to turn"), 3- (corresponding to the property "unemployed") and others are located in the sentence, which begins from the 2nd line of the text of the document and they occupy the place from the 15-th to the 310-th byte. These means of positioning are necessary for the work of the reverse linguistic processor (LP).

Analyzing this example, it is possible to make the following conclusions: 1) In SS- document the estimating (characterizing) words occur either in one fragment with the object - SCANDAL (,…), or the next one, i.e., the codes of the actions, in which the object participates, are nearby in PREDL_(… 9, 13-, 14…). In this case the possibility of composite actions is considered. 2) On the actions, represented as SCANDAL (,…), it is possible to

draw the conclusion that the discussion deals with "that suspected", and TO REPORT (,) - that the person is "suffered" or "the applicant". Such conclusions are easily arrived at with the aid of the rules IF… THEN (productions) of the language DEKL, which are the basis for the extraction of role functions. 3) The particular difficulties of dividing the text into the sentences occur (in the old version). The reduction "of n/r" (with the point at the end) was not understood as the end of a sentence. 4) The linguistic processor (LP) correctly identified the pronoun "*he*", and also it knew how to reveal the participation of the subject ("*unknowns*") by the actions "*to be evinced by unquotable swearing*" and "*to set dog*", which also characterize subject. At the same time the LP could not connect the action "*diagnosis was set*" with the person - "*Mitrofanov*…" (the code is 2-nd). In this case an example proved to be successful. Also the processor LP (with its linguistic knowledge - LK) was developed for the tasks of the criminal police, connected with different forms of the objective searches: the search for similar pparticipants (addresses, and so forth), search according to the connections, precise search for objects, for the search by signs and other identifiers. In this case the analysis of some complex NL forms was not required, i.e. the cases of the enumeration of the objects participating in the uniform actions (they are described by one verb), the enumeration of the actions of one object and others in contrast to the aforesaid, with the extraction of role functions for each object the indication of its participation in each action is required. Hence it follows that with the use of the proposed procedure the more qualitative extraction of role functions is directly connected with the works on improvement of LP in the aspect of the development of objects and their actions. In many instances the numerous errors caused the inaccuracies in SS-document, e.g.: the absence of punctuation marks or their presence (where it was not required), the inappropriate reductions, gaps in the words and many others. The fact is that the documents, entering the summaries of incidents, are composed on the spot by people (militiamen) of different degree of literacy. Hence – the additional noise and loss. Thus, meaningful portraits are the collections of fragments of ESN which represent the sufficiently high level of formalization of NL texts and are convenient for the working - with the aid of the instrument means - DEKL [3]. Besides LP which analyzes texts and builds SS-documents, there is a reverse linguistic processor (LP) which on the basis of the fragments of the SS- document generates the NL texts presented to the user [6].

4 The means for the establishment of the role functions

As it has already been said, within the framework of the proposed procedure (instead of the application of syntactical-semantic forms to the documents) the rules are used for logical conclusion and transformation of the knowledge structures - the SS- documents, in which there are no morphological features (of type who, whom,…), and the subjects and the objects are distinguished by their arrangement in the fragments of ESN, which present actions. The names of fragments present the nature of actions. Syntactical-semantic forms are transformed into the fragments of ESN which determine conversions and logical conclusion achieved by productions of language DEKL. Such fragments play the role of the logical-semantic shell, which determines conversions and logical conclusion on the basis of SS-documents. After filling of the shell by ontological-fragmental knowledge (OFK) which consist of the mentioned fragments (ESN), the program is formed, which accomplishes the development of role functions and completion of the SS-document by the appropriate fragments. With this approach it is possible to avoid many difficulties, connected with the design features of NL and the specific character of the use of syntactical-semantic forms. There are many versions of construction of the shells and representation of the corresponding knowledge which are distinguished according to the degree of their generality. Let us examine the version which is at present realized and verified.

Case 1. The role functions are determined by the names of actions. In this case for the extraction of objects (participants) which should be assigned properties (role functions), the fragments of the following form are used :

INTERPRET (MAN_2, FIO, " suffered") FORMA_CC (MAN_2, CLASS_D4, " ") CLASS_D4 (TO TURN, TO STATE, TO REPORT, TO PASS AWAY,…)

The first fragment INTERPRET (…) means that from the SS- document it is necessary to extract the fragments of the form FIO (…), that correspond to participants, and to analyze the possibility of assigning them the property "suffered". Such participants are conditionally designated as MAN_2. The second fragment FORMA_CC (…) specifies the conditions for assigning this property to MAN_2, determined by the constant CLASS_D4. In the third fragment CLASS_D3 (…) the words are given which present actions. It is represented that the words belong to the class CLASS_D3. If the participant occurs in one of the enumerated actions, then to this participant the property "suffered" is assigned. This participation is revealed via the analysis of the SS-document. If there is a fragment TO TURN (…, it is n-th,…) in it, the argument of which is the code FIO (… /N+), then the fragment N-("suffered") is added that represents the role function of the corresponding participant. Conformably for the SS-document represented in example 2 the analysis will occur as follows. Consecutively the extraction of fragments FIO (…) corresponding to the participants is performed. First FIO (MITROFANOV,… /2+) will be extracted . Its code is 2- is the argument of the fragment TO TURN (1, GR- N, 2-/11+), that presents the action. In connection with this to SS- document the fragment 11-("suffered") will be added, which via the reverse LP will be transformed into the statement that "Mitrofanov Victor Mikhaylovich is a suffered person". These actions are realized within the framework of the logical-linguistic shell.

Case 2. Role functions are determined by the actions and elucidating words. For this the same fragments are

used, as in the first case, but during the enumeration of the names of actions the additional fragments which present actions with the possible elucidating words, are introduced:

INTERPRET (MAN_1, FIO, "suspect")
FORMA_CC (MAN_1, CLASS_D3, " ")
FRAUD (USER, POKUPATEL/15+)
TO SET (SOBAKA/16+)
TO BE EXPRESSED (UNQUOTABLE, SWEARING, MATERNYY,… /17+)
CLASS_D3 (IS DELAYED, TO BE SOUGHT,…, 15,16-, 17-)

The given fragments determine actions of the extraction of persons (MAN_1), by which the property of "suspect" is assigned. For this at the level of the knowledge structures their participation is analyzed in the actions "*is delayed*", "*to be sought*", and also in the composite actions: "*to set dog*", "*to be expressed unquotable…*", "*to be expressed by swearing…"* and others. In example 2 the code of fragment FIO (" " , " " , " " , NESKOLKO]6+), that represents the unknown persons is the argument of the fragment TO SET (6, SOBAKA/14+), representing action *"to set"* with the elucidating word *"dog" – "sobaka"*. Therefore the fragment 6 is added ("*suspect*"), that represents that "*the unknown persons are suspected*", and through the reverse LP the explanation to this conclusion is offered, see below. A similar conclusion will be made on the basis of the fragment TO BE EXPRESSED (6, UNQUOTABLE, BRAN/13+), but with other explanations.

Case 3. The actions determine the role functions of several persons. For this (additionally to the fragments INTERPRET) the fragments are added: CLASS_D1 (TO STRIKE, TO BEAT UP,…) FORMA_CC (MAN_1, CLASS_D1, MAN_2), where FORMA_CC (…) indicates the need of the search of two persons - "*suspect*" and "*suffered*" (MAN_1 and MAN_2), that participate in one action, which are mentioned in the fragment CLASS_D1 (…). For example, "*certain person struck another…*". In the appropriate fragment TO STRIKE (…) the code FIO (…) that corresponds to the first person will stand in front of the second. The given fragments of ESN compose the knowledge OFK which are constantly supplemented - due to the filling of classes by the new words-actions and with the elucidating words. The process of filling is sufficiently simple. If role function is not revealed, then it is necessary to look in the SS- document in which the action of one or another participant (by the text its role is easily determined) occurs. Further, the corresponding constants are located, by which are supplemented the classes of knowledge OFK. Subsequently it is intended to automate the process of completing the knowledge OFK as follows. In the text the words, which determine role functions, are noted. Further, in the formed SS-document the corresponding constants which supplement knowledge OFK are located.

5 Explanation of the results

The explanation of results is accomplished through the reverse LP which on the basis of SS- document and additional fragments builds texts in natural language which are displayed to the user. The reverse LP through the codes of the fragments which correspond to object and actions, finds the sentence (PREDL]) and its location in the text of a document. Through the arguments of these fragments (the words in the normal form) the processor finds the components of the sentences in which the mentioned object and actions are described. These components are converted into the form suitable for the delivery to the user. The fact is that many components are transformed depending on the context. For example, "*… he threatened Petrov I.I.…..*" – "*…ugrozhal Petrovu I.I….*" , where during the delivery FIO "*Petrovu"* should be transformed into "*Petrov I.I.*" Further the description of the object is delivered, its generated property and actions which explain this property - role function.

Example 3. with the use of the above given knowledge OFK of the SS - document of example 1 the role functions will be generated which with the aid of the reverse LP will be given out to user in the following form:

unknowns - suspected,
since - unknowns, being in the drunk state made scandal,
they expressing themselves by unquotable swearing, they set dog
Mitrofanov Eugene Mikhaylovich, 1953 r. - suffered,
since – applied to OVD citizen Mitrofanov Eugene Mikhaylovich, 1953 yr. birth,
since Mitrofanov applied to trauma care centre.

It should be noted that all the actions considered in the paper connected with various cases of role functions establishment and explanations of the results are implemented within the framework of the logical semantic environment written in the logical programming language DEKL. Since the DEKL language is oriented at the processing of knowledge structures (represented in the form of the extended semantic networks – ESN) and since it features the generalized production rules [6], the program code in the DEKL language is very simple and concise: it comprises 16 productions and about 4 Kbyte of text.

6 Conclusions

The proposed procedure of the role functions extraction centered at the analysis of knowledge structures is sufficiently promising from the point of view of the knowledge bases technology development. The current task is to improve its performance for the documents comprising enumeration of the type: 1. *Ivanov I.I.…* 2. *Petrov A.A.….* 3.… and further follows the continuation, which describes their acts, for example, "*were subjected to detention by…*" or "*who performed…*". The recognition of such cases requires further upgrade of the linguistic processor (LP) software. The quality of analysis is lowered by the breaks in significant words of the type "*Iva nov*" or "*Iva-nov*", which are typical for the summaries of incidents. The methods were tested on the basis of the summaries of

incidents which contain about three thousand documents (each document consists of 10 - 80 lines). In the case of the summaries processing the documents with the mentioned enumerations (there were about 10% of them) were withdrawn and in the remained texts the gaps in the words were removed. At the current moment the program which realizes the proposed procedure gave about 80% of correct recognition of role functions, and about 65% of complete explanations with the indication of all acts. But these numbers rapidly change for the better due to the means (the LK and OFK knowledge) of tuning the LP to special features of the subject area texts. For this not much time is required. Let us note that tuning itself to the extraction of the role functions of persons from the mentioned summaries (with reaching the indicated percentages), required about two weeks of the work of one person. The development and fixing of the shell itself took about four days. The subsequent development is connected with the improvement and the tuning of LP to the work with complex NL forms. At present the extraction of actions is interfered with causal word combinations of the type "*out of the hooligan motives*", "*owing to the hostile relations*" and so forth, which at present are introduced into the system. Difficulties appear with the transfer of the subject of action to other actions to which the subject is not assigned explicitly, but its presence is implied.

The second direction of research and development is connected with the extension of the shell features to the solution of other problems connected with the estimation of objects depending on the nature of statements about them in the texts of description. Within the framework of the studies conducted it is also intended to tune the shell to the work with the English language texts. Since the meaningful portraits of the English language and Russian language texts have the identical structure (SS-documents), this tuning cannot be labor-consuming.

References

[1] Кузнецов И.П. Семантические представления // М. Наука. 1986г. 290 с.

[2] Igor Kuznetsov, Elena Kozerenko. The system for extracting semantic information from natural language texts // Proceeding of International Conference on Machine Learning. MLMTA-03, Las Vegas US, 23-26 June 2003 г., p. 75-80.

[3] Кузнецов И.П. Методы обработки сводок с выделением особенностей фигурантов и происшествий // Труды международного семинара Диалог-1999 по компьютерной лингвистике и ее приложениям. Том 2. Таруса 1999.

[4] Кузнецов И.П., Мацкевич А.Г. Семантико-ориентированный лингвистический процессор для автоматической формализации автобиографических данных. Труды международной конференции по компьютерной лингвистике и интеллектуальным технологиям "Диалог 2006", Бекасово, 2006, стр. 317-322.

[5] Кузнецов И.П., Ефимов Д.А. Особенности извлечения знаний семантико-ориентированным лингвистическим процессором Semantix.// Сб. Компьютерная лингвистика и интеллектуальные технологии. Выпуск 7 (14). По материалам конференции «Диалог 008»..РГГУ, М.:2008., С. 281-291.

[6] Кузнецов И.П., Мацкевич А.Г. Семантико-ориентированные системы на основе Баз Знаний.// Монография, МТУСИ. М.: 2007. 173 с.

[7] Asher, N. & Lascarides, A. *Logics of conversation.* Cambridge etc.: Cambridge university press, 2003.

[8] Кузнецов И.П., Сомин Н.В. Средства настройки семантико-ориентированного лингвистического процессора на выделение и поиск объектов. Сб. ИПИ РАН, Вып.18. 2008 г., стр. 119-143 .

[9] Kuznetsov I.P., Kozerenko E.B. Linguistic Processor "Semantix" for Knowledge extraction from natural texts in Russia and English. Proceeding of International Conference on Machine Learning, ISAT-2008. 14-18 July, 2008 Las Vegas, USA// CSREA Press, 2008, p.835-841.

[10] Кузнецов И.П., Мацкевич А.Г. Англоязычная версия системы автоматического выявления значимой информации из текстов естественного языка // Труды международной конференции по компьютерной лингвистике и интеллектуальным технологиям "Диалог 2005", Звенигород, 2005.

[11] Banko M., M. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open Information Extraction from the Web // Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07), 2007. P. 2670–2676.

[12] Clark P., P. Harrison, and J. Thompson. A Knowledge-Driven Approach to Text Meaning Processing // Proceedings of the HLT-NAACL 2003 Workshop on Text Meaning, 2007. P. 1–6.

[13] Gildea D. and M. Palmer. The necessity of syntactic parsing for predicate argument recognition. In Proceedings of the 40th Annual Conference of the Association for Computational Linguistics (ACL-02), Philadelphia, PA, 2002. P. 239–246.

[14] Pa¸sca M. and B. Van Durme. What You Seek is What You Get: Extraction of Class Attributes from Query Logs // Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07), 2007. P. 2832–2837.

[15] Punyakanok V., D. Roth, and W. tau Yih. The Importance of Syntactic Parsing and Inference in Semantic Role Labeling // Computational Linguistics 34(2), 2008. P. 257–287.

Set-Phrase Machine Translation Based on Multilingual Dictionaries

Alexander Khoroshilov and Alexei Khoroshilov

Institute for Informatics Problems of the Russian Academy of Sciences, Moscow, Russia

Abstract. *The paper focuses on the issues of automatic compiling of the set-phrase dictionaries for machine translation systems. The methods employed are based on translation memory acquisition principles and heuristic language processing tools. Machine learning techniques are used for extraction of new rules and templates.*

Keywords: machine translation, set-phrase dictionaries, machine learning, translation memory

1 Introduction

The systems of machine translation of texts simulate operation of a human translator . Their efficiency depends on how the nature of language operation and mentation are taken into account, and this nature has not yet been adequately studied. Therefore, developers, attacking the problem of machine translation, must take into account the experience in international communication and translation activity, that was accumulated by mankind. That experience testifies that in the process of text translation the phraseological word combinations expressing the concepts rather than single words are the basic units of sense. The concepts are the elementary intellectual images, by the use of which it is possible to create more complex intellectual image corresponding to translated text.

The mankind have been concerned with the problem of machine translation already for more than half a century. But until recent time, the progress in that area of research and development has been insignificant. The effectiveness of created machine translation systems was not high. A reason for this situation is the primordial orientation of the pioneers of machine translation and their followers on the semantic-syntactic and mainly word-by-word translation. In this case, the single words were considered to be basic units of sense expressing the concepts, and the sense of larger speech units (word combinations, phrases and utterance-length units) was supposed to be determined on the base of the sense of words comprising them. In dictionaries the use of word combinations along with single words was also admitted. But these combinations were mainly the idiomatic expressions, and their amount in dictionaries of machine translation systems was negligible in comparison with the amount of single words .

2 Concept of phraseological machine translation

The orientation on the semantic-syntactic and mainly word-by-word translation couldn't lead to the solution of the fundamental problems of machine translation, because within language and speech the sense of units of higher level, as a rule, cannot be reduced or fully reduced to the sense of the lower level units comprising them. Almost all known systems related to traditional machine translation systems, developed in that direction. Later the developers of traditional systems began to include more terminological word combinations into their dictionaries.

The concept of translation memory (sentence memory) appeared as an alternative to traditional machine translation. That concept can be regarded as an attempt to realize the idea of Japanese computer scientist Makoto Nagao, that in the process of machine translation it is necessary to use the large corpora of parallel texts, earlier translated by humans. A more adequate approach is based on the concept of statistical machine translation (statistical-based machine translation), which is defined by some authors as a "sort of machine translation of texts, based on comparing of large corpora of language pairs". In contrast to traditional machine translation, statistical approach is based on statistical computation of matching probability and doesn't use the linguistic algorithms. Large corpora of parallel texts are necessary for operation of this system. A statistical mechanism of text analysis is used in the process of translation. This mechanism allows to select the variant for the word combination translation based on matching frequency of the language pair elements. A weak point of statistical systems is partial or total absence of a mechanism of grammatical rules analysis for source and target languages. Therefore, it is hard to imagine that the system, which doesn't analyze the text from the point of view of grammar, is able to release the correct translation of semantically complex texts.

The idea to create machine translation systems on the base of previously translated texts presented in the form of "large corpora of language pairs" is not objectionable. It can be realized in different ways. The first way provides text translation with the use of statistic analysis of large corpora of bilingual texts in the process of translation. This way is "statistic machine translation". The second way is connected with a single-step

compiling of bilingual frequency dictionaries of words and phraseological word combinations. The creators of the systems of phraseological machine translations follow the second way. This way excludes the fatal dependence of the translation process on availability of large volumes of parallel texts and quality of their translation.

The systems of phraseological machine translation are based on the theoretical concept, the main thesis of which is a statement that the concept names in texts are determined by word combinations rather than single words. Therefore, in the process of text translation from one language to another, it is necessary to use the phraseological combinations expressing the concepts, relationships between concepts and the typical situations rather than single words as basic units of sense. The single words may also be used if the translation with the help of the phraseological word combinations fails.

In compliance with that thesis, the system of phraseological machine translation must comprise the knowledge base of translation equivalents for most frequent phrases, phraseological combinations and single words. In the process of text translation the system should use the translation equivalents stored in its knowledge base in the following order: at first, an attempt to translate the successive sentence of the source text as the integral phraseological unit is made; then, if this attempt fails, words combinations being a part of the sentence should be translated; and, finally, if both above-mentioned attempts fail, word-by-word translation of the text fragments is performed. The fragments of the target text translated with the use of all three approaches, must grammatically agree with one another with the help of procedures of morphological and syntactic synthesis). Let's give consideration to this conception in detail. It is necessary to apply the following principles in the process of development of phraseological machine translation systems:

1. The phraseological units (word combinations and phrases).are basic language and speech units, which should be primarily included in the computerized dictionary.

2. Along with the phraseological units composed of continual word sequences, so called "speech models" - phraseological units with blank spaces, that may be filled with different words and word combinations, generating meaningful segments of speech, may be used in machine translation systems.

3. Real texts, without regard to their subject area, tend to be polythematic, if they have sufficiently large size. These texts differ from each other not so much by word stock as by probability distribution of occurrence of different words and word combinations from national word stock in them. Therefore, the computerized dictionary designed for translation of the text belonging to a single subject area must be polithematic, not to speak of translation of texts belonging to different subject areas.

4. Systems of phraseological translation need high-volume computerized dictionaries. Such dictionaries should be created on the base of computer-aided processing of parallel texts - bilingual texts, which are translations of each other, and in the process of translation system operation .

5. Along with the main high-volume polithematic dictionary, it is also reasonable to use a set of additional small-volume highly specialized dictionaries in systems of phraseological machine translation. The additional dictionaries should only contain information missing from the main dictionary (for example, data on priority translation equivalents of word combinations and words for different subject areas, if these equivalents are not equal to priority translation equivalents of the main dictionary).

6. The main means for solution of the problem of words polysemy in phraseological translation systems is their use in phraseological word combinations. The additional means is a set of additional specialized dictionaries , where the priority translation equivalent specific for subject area in question is identified for each multiple-meaning word or word combination.

7. The procedures of morphological and syntactic analysis and synthesis of texts, that are built on the base of linguistic analogy may play a major role in the systems of phraseological machine translation of texts.. These procedures allow to give up storing large amounts of grammatical information in dictionaries and generate it automatically in the process of translation when the need arises. They make the translation system open - able to process texts with "new" words.

8. Along with text translation in automatic mode, it is reasonable to provide for an interactive mode of operation for the systems of phraseological machine translation. In that mode the user should have potentiality to intervene in the translation process and adapt the additional computerized dictionaries for the subject area of the translated text.

3 Digital dictionaries

First of all, the systems of phraseological machine translation should be orientated on translation of texts on business, science, technologies, politics and economy. Translation of literary texts is- more complex task. But success can be also achieved in this area in future, if modern technological means are used to compile huge phraseological dictionaries for these texts.

At first glance, the machine translation concept offered by professor Makoto Nagao in 1984, fundamentally differs from the concept, formulated by professor G. G. Belonogov nine years earlier. . But this is not true. Indeed, in the process of practical realization of Makoto Nagao's concept, it is difficult to imagine that the text written in any language is completely the same as another text written earlier and translated into foreign language. It is not to be expected that this text contains long fragments (chapters, paragraphs and etc.), that are the same as the fragments of the text written and translated earlier But, as our investigations showed, the continuous texts fragments including over ten words repeat on rare occasions - their total frequency doesn't exceed 1%. It is necessary to use only short sentences, single words and text fragments (word combinations)

including less than 10-12 words. This is the semantic-syntactic phraseological translation.

Of course, along with the translation equivalents of the relatively short fragments of texts, it is possible to include the translation equivalents of longer fragments in the computerized dictionaries. But in this case one should keep in mind, that the computerized dictionaries will be filled with "dead" ballast, i.e. with the dictionary entries, which will be used on rare occasions or will not be used at all in the process of text translation.

When developing the systems of phraseological machine translation, most difficult and time-consuming problem is a problem of compiling sufficiently high-volume computerized dictionaries. The quality of translation depends on the volume of these dictionaries and on the quantity of the phraseological word combinations in them. And those volumes needs to be sufficiently large to provide the good covering of texts.

It is known, that in modern languages of the world (for example, in Russian or English) the amount of different words exceeds one million, and the amount of concept names determined by word combinations exceeds hundreds of millions. The authors of this article came to this conclusion on the basis of many years' experience of the statistical study of texts. Confirmation of such viewpoint is the report of All-european terminological centre "Infoterm" (Vienna, Austria, 1998), in which it was found that in modern languages of the world, such as English and German, a total amount of different terms exceeds 50 million, and nomenclature of goods exceeds 100 million. It is well known, that the connected texts consist of not only terms and names of goods.

The computerized dictionaries of such volume cannot be created quickly, but as experience shows, it is possible to achieve satisfactory quality of translation at the first stage in the presence of only several millions of entries in dictionaries, at least 80% of which should be word combinations. In this case the polithematic texts have the coverage of about 99, 7%.

Thereafter, the volume of dictionaries must be constantly increased and with the growth in amount of phraseological combinations, the quality of machine translation should improve. This problem cannot be solved by manual methods. For its solution, a system of computerized compiling and maintenance of the computerized dictionaries was created.

4 System operation

The implementation of the computerized phraseological text translation from one language into another must have three stages. An first stage, the semantic-syntactic analysis of the source text is carried out. During that analysis the text is splitted into sentences and then their conceptual and syntactic structure is determined . At the second stage (at the transfer stage) the concept names of the source text are substituted by the concept names in target language and the information on the syntactic structure of the source text is transformed into information required for the target text synthesis. At the final stage (the stage of semantic-syntactic synthesis of the target text) the text in the target language is formed.

The stages listed above are present in the process of translation of texts from any language to any other language, but their specific content for different pairs of languages has a specific character. This specific character can be seen in procedures of semantic-syntactic analysis and synthesis of texts, which include the procedures of morphological, syntactic and conceptual analysis and synthesis [1-6].

The set-phrase machine translation based on the multilingual dictionaries will operate in the same way, but these systems should be complemented with the procedures of semantic-syntactic and conceptual analysis and synthesis of all languages, which will be included in the system. The authors developed the effective technology based on the use of principles of linguistic analogy for creation of these procedures.

The computerized dictionaries are the most important part of the systems of phraseological machine translation. They should have sufficiently large volume, in order to cover texts, and should contain mainly word combinations. The authors developed the original methods, algorithms and programs for automated compiling and maintaining dictionaries for the system of phraseological machine translation. In cooperation with other specialists, the large-volume Russian-English and English-Russian phraseological computerized dictionaries containing 2, 6 million dictionary entries each were compiled. These dictionaries cover 99, 7% of the lexical content of modern texts and they represent the -powerful bilingual conceptual model for a wide range of fields of human activity .

5 Automation of dictionary compiling

The experience in creation of the large-volume Russian-English and English-Russian computerized dictionaries convinced authors, that Russian and English texts, which are translations to each other (for example, bilingual titles of the documents), can serve as the most reliable source for dictionaries compiling.

The compiling of the computerized dictionaries with the use of bilingual texts was carried out both manually and with the assistance of computers. The manual dictionary making requires huge expenditures of human labour. Therefore the authors of the article developed the procedure for automated dictionary making [3].This procedure is based on the hypothesis, that in numerous bilingual pairs of sentences, which are translations to each other and which contain the same word or word combination of one of the languages, the word or word combination of another language, which is the translation of this word or word combination has maximal occurence frequency.

The procedure was used for processing bilingual (Russian and English) titles of the documents from the databases of VINITI (All-Union Scientific and Technical Information Institute). In this case more than one million pairs of the document titles were processed.

The computerized dictionaries of RETRANS system can be corrected and completed in the process of text translation in the interactive mode. In that mode there is an opportunity to identify the words and word combinations, which have no translation equivalents in the dictionary or these equivalents do not comply with the context or several equivalents are given, but the first equivalent does not comply with the context. These equivalents can be replaced by the equivalents complying with the textual context .

In compliance with the method described above, the large-scale experiment on compiling English-Russian frequency dictionaries on the base of the automated concept analysis of English and Russian titles of the documents, which are translations to each other, was carried out. For this purpose, the corpus of English titles of the polithematical documents and their Russian translations having the volume of about 2 million pairs of sentences from the VINITI's databases (1994-1999) were processed. The total volume of the corpus of texts is 390 Mb.

In the process of research three English-Russian frequency dictionaries were created:

1. The dictionary, items of which are the combinations of fragments of English and Russian titles of documents, between which the translation equivalents were determined with the assistance of RETRANS system;

2. The dictionary, items of which are fragments of titles of documents, between which the translation equivalents were not determined, but they are surrounded by the other fragments, between which such equivalents were determined , or by signs of beginning or end-of-the title;

3. The dictionary, items of which are fragments of English and Russian titles of documents, between which translation equivalents were determined on the initial stage of titles processing.

The first frequency dictionary includes bilingual phraseological word combinations containing 2 to 16 words. It had 3.127.363 dictionary entries.

The value of the dictionary in question is that it contains translation equivalents between English and Russian fragments of titles of documents, which are longer than their fragments selected at the first stage of conceptual analysis of titles. Each of the newly formed dictionary entries practically has just one translation version of English word combination (the percentage of dictionary entries having more than one version of translation is less than 0.1).

The second frequency dictionary includes translation equivalents between fragments of titles of documents, that were not found at the initial stage of conceptual analysis of these titles. This dictionary contains 1.825.612 dictionary entries. The most frequent dictionary entries have the frequency of 1.008, and infrequent dictionary entries have- the frequency equal to one. 87% of dictionary entries have the frequency equal to one. A spot-check of the dictionary showed that about 50 % of translation equivalents were incorrect. The quantity of such translation equivalents can be reduced at the final stage of dictionary making, if the procedure of semantic-syntactic checking is applied. After that, the dictionary must be edited by humans.

The third frequency dictionary contains translation equivalents between fragments of English and Russian titles of documents that were found at the initial stage of processing of these titles. It contains 387.025 dictionary entries. The most frequent dictionary entries have the frequency of 4.985, and most infrequent dictionary entries have- the frequency equal to one. 56% of dictionary entries had the frequency equal to one.

This dictionary contains only translation equivalents of English concept names, which are found in Russian titles of documents. Therefore, it proved to be tuned to the subject field of translated titles. It allows to consider the compiling procedure of such a dictionary as a means for automated creation of thematically oriented dictionaries.

6 Conclusion

In conclusion, it should be noted that this article describes the experience in creation of modern multilingual machine translation systems – the systems of phraseological translation. The extensive application of means of automation allowed to essentially reduce expenditures of human labour in the process of creation of this system, and therefore, to reduce the creation cost of such systems.

References

1. Belonogov, G. G., Khoroshilov, Alexander A., Khoroshilov, Alexei. A. Phraseological Machine Translation of Texts from Natural Languages to Other Natural Languages. . Col **:**"Scientific-Technical Information", Series 2. - M.: VINITI, 2010, № 10.

2. Belonogov, G. G., Khoroshilov, Alexander A., Khoroshilov, Alexei A. **A**utomatization of compiling of English-Russian bilingual phraseological dictionary using the corpora of bilingual texts. Col.**: "**Scientific Technical Information**"**, Series 2. - M **,** VINITI, 2010, № 5.

3. Belonogov, G. G., Gilyarevskij, R. S., Khoroshilov, A.A. **O**n the nature of information. Col.**:**"Scientific Technical Information**"**, **S**eries 2. - 2009. - № 1.

4. Belonogov, G. G., Kalinin, Yu. P., Khoroshilov, Alexander A., Khoroshilov, Alexei A.. **S**ystems of Phraseological Machine Translation of Texts. Theoretical Preconditions and Experience in the Development. - M. 200**7.**

5. Belonogov, G. G., Kalinin, Yu. P., Khoroshilov, A. A. **C**omputational linguistics and Advanced Information Technologies. Theory and Practice of Constructing of Automatic Text Processing Systems. - M. 200**4**.

6. Belonogov, G. G., Bystrov, I. I., Kozachuk, M V., Novoselov, A. P., Khoroshilov A.A. Automated Conceptual Text Analysis. Col. :**"** Scientific Technical Information**"**, **S**eries 2. - 2002. № 10.

Sana'ani Dialect to Modern Standard Arabic:
Rule-based Direct Machine Translation

Yahya Alamlahi *yahya.alamlahi@hotmail.com*, and **Fateh Ahmed** *al_shuaibifth@yahoo.com*,
Computer Science Dep., Sana'a University, Sana'a, Yemen, 2007

Abstract - *This paper presents an algorithm to normalize Sana'ani dialect to modern standard Arabic. The algorithm based on morphological rules related to Sana'ani dialect as well as Modern Standard Arabic. Such rules facilitate the dialect conversion to its corresponding MSA. The algorithm tokenizes the input dialect text and analyses each token into stem and its affixes; such affixes can be categorized into two categories: dialect affixes and/or MSA affixes. At the same time, the stem could be dialect stem or MSA stem. Therefore, our algorithm must pay attention to such situations. That is done by using of a simple MSA stemmer. Then our dialect stemmer is applied to strip the resulting token and extract dialect affixes. At this point, the rules take place to decide when to carry out the extraction of an affix. The experiment shows that Sana'ani dialect has three classes of distortions. They are prefixes, suffixes, and stems distortions. The algorithm normalizes such distortion based on the morphological rules. For each morphological rule the algorithm checks possibility of applying such rule. That means if rule conditions meet, then the dialect affix will be replaced by its MSA correspondence. If there is no restriction on applying the rule related to the distorted stem, then the rule can be considered as a parallel corpus of dialect and MSA. Finally, the experiment computes the distortion ratio of MSA in Sana'ani dialect. For Sana'ani dialect sample of 529 words, 6.98% of them have distorted suffixes, 5.65% have distorted prefixes and 11.72% contain distorted stems. These percentages are related to processed words only.*

Keywords: Modern Standard Arabic, Yemeni, Sana'a Dialect, Rule-based Machine translation, MSA, NLP

1 Why Arabic Dialects?

Arabic is garnering attention in the NLP community due its linguistic difference from Indo-European languages. Modern Standard Arabic (MSA) is a form of Arabic language that is used widely in news media and formal speeches. There are no native speakers of MSA. The importance of processing dialect comes from here. "Almost no native speakers of Arabic sustain continuous spontaneous production of MSA. Dialects are the primary form of Arabic used in all unscripted spoken genres: conversational, talk shows, interviews, etc. Dialects are increasingly in use in new written media (newsgroups, weblogs, etc.). Substantial Dialect-MSA differences impede direct application of MSA NLP tools." [Diab, Habash, Ch1.Ref.8]. Converting Sana'ani dialect to MSA enables MSA NLP tools to process this dialect indirectly. For example, it is easy to parse the translated dialect using MSA parser rather than developing a special parser for that dialect; in this example, suboptimal MSA output can still be helpful for the parsing task without necessarily being fluent or accurate since the goal is parsing [Parsing Arabic Dialects, Ref.4]. Such these fields of researches suffer from lack of resources due to lack of standards for the dialects and lack of written resources of dialects themselves. Dialect to MSA translation is useful in many other applications, such as querying dialect text using MSA words; search engines should retrieve appropriate results from dialect pages as well as MSA pages with the same query that is written usually in MSA.

2 Implementing Dialect-MSA variations

Table 1 Rules Sample

Rules Name	Rule Type	Previous word	Previous word type	Next word	Next word type	MSA equivalent
ـش	Enclitic	ما	Null	Null	Null	Empty string
	Enclitic	Null	Null	Null	Null	كِ (feminine, 2nd person)
عـ	Proclitic	Null	Null	Null	Verb,Muthare3	سـ
مش	Stem	Null	Null	Null	Null	ليس، ما

Our implementation for rule can be applied to any distortion in MSA language in Sana'ani dialect. Rule depends on pattern length of two words plus the word that is distorted and falls in focus. The two words may be any words of MSA or two word types. Matching is done with a pattern of sentence. Processing pattern of dialect is a sentence. Table 1 shows three rules as an example.

For dialect clitic that has more than one rule, the rules must be ordered from the more specific to less specific. 'Previous' stands for any appearance of word in previous words with

respect to the word that has distortion (the word in focus) within the sentence.

Here, we don't pay attention to grammar and fluency consideration. It can be processed in other phase. In example #1 and #2 (table 2), different rule is applied to the same enclitic 'ش'. If the rules are not prioritized, there is no guarantee to apply rule 'empty string' of 'ش' at all, because of the other will be applicable all time. To apply rule we should use a stemmer to separate token into smaller ones. We don't need a complex stemmer. Stemmer, that is capable to stem token in parts these are understood by rules, is enough.

Table 2 Sample of translation

No#	Ex#1	Ex#2	Ex#3
Sana'ani dialect	ما قدرتـ**ش** العب معهم mA qdrt_**\$** AlEb mEhm	لعبـ**ش** **مش** هو حالي lEb_**\$ m\$** hw HAly	ما **عـ**تقدر**ش** تلعب معهم mA **E**__tqdr**\$** tlEb mEhm
MSA: Distortion correction	ما قدرت ألعب معهم mA qdrt >lEb mEhm	لعبك ما هو حالي lEbki mA hw HAly	ما ستقدر تلعب معهم mA stqdr tlEb mEhm
MSA: Grammar and fluency consideration	ما أستطعت أن العب (اللعب) معم mA >stTEt >n AlEb (AllEb) mEm	لعبك ليس حسن lEbki lys Hsn	لن تقدر(تستطيع) ان تلعب معهم ln tqdr(tstTyE) An tlEb mEhm
Distortion type	Enclitic distortion at '**ش**ـ' in '**قدرتـش**'	Enclitic distortion at '**ش**ـ' in '**لعبـش**' and stem distortion in 'مش'	Proclitic distortion at '**عـ**' and enclitic distortion at '**ش**' in '**عتقدرش**'

3 Stemming

Before applying the rule, we must provide the rule with an outcome of stemmer. This is important because the Sana'ani dialect contains MSA clitics, dialect clitics with MSA word or combination of both. Each of them must be divided apart to be easily processed by rules. We borrowed our stemmer idea from ISRI stemmer which is automatic stemmer without root **dictionary**.

Rules that need to specify word types can be solved by using an **automatic tagging** of words in Arabic language or by using a **dictionary** contains MSA stem along with their types. A dictionary with words, those have types mentioned in rules, is enough fairly. In fact, we will restrict our research to this typed-dictionary.

Table 3 MSA clitics sets

Set	Description	Examples
D	Diacritics-vowelizations	'ـِ','ـُ','ـَ','ـّ','ـْ','ـٍ','ـٌ','ـً'
P3	Prefixes of length three	"وللل", "وال", "كال", "بال"
P2	Prefixes of length two	"ال", "لل"
S3	suffixes of length three	"كما", "تين", "تان", "هما", "تما"
S2	suffixes of length two	"ين", "كم", "هن", "نا", "ون", "ات", "ان", "ني", "وا", "ما", "هم", "يا", "ها", "تم", "كن", "تن"

4 Translation Algorithm

The following steps describe the algorithm and utilize the above technique:

1) Remove diacritics representing vowels. Set D on table 3.
2) Remove connector و if it precedes a word beginning with و.
3) Remove length three S3 and length two S2 suffixes of MSA in that order. Extract them from the token. MSA suffixes are shown in table 3.
4) Remove length three P3 and length two P2 prefixes of MSA in that order. Extract them from the token. MSA prefixes are shown in table 3.
5) Replace suffixes of dialect with their MSA alternatives according the rules whenever their conditions are met. Process the longer prefixes first then shorter ones. Extract them from the token.
6) Do the same (step 5) with prefixes of dialect.
7) Extracting stem by removing dialect clitics regardless of applicable of rule may resolve the dependency of rule. Rule may not be applied until next word processed. Next word also may need processing of the previous one. In general, our rules haven't such this deeper dependency except in distorted MSA stems those have dialect clitics.
8) Check the remaining token. If it is a dialect stem, apply stem rules and get the alternative MSA stem.
9) Rebuild the token by adding the removed MSA clitics and dialect repaired clitics to the stem.
10) Unrecognized token are unchanged.

5 Experiment

We have applied the algorithm on a sample of Sana'ani dialect contains 529 words; we get the results shown on table

4. These numbers are restricted to our application and related to processed words only. The results are governed by the total data stored in the application about the dialect-MSA differences, such as ratio of stored rules to the all possible rules, and the size of parallel corpus as well as stored tagged words.

Other good experiment we made is using our approach as mid-layer for Sakhr Text Mining engine which is developed for Arabic. We present Sana'ani dialect sample and its MSA translation to Sakhr Text Mining Engine and we find that the mining is enhanced greatly when using our approach as mid-layer. Sana'ani Dialect → our Dialect/MSA Translator → Sakhr Text Mining Engine → Analysis Results.
The results are changed noticeably. The accuracy is increased. The document classification is defined well. Keywords are extracted successfully. This result was not possible when using Sakhr text mining engine alone.

Table 4 Experiment Results

Total words	529
Processed MSA suffixes	80
Processed MSA prefixes	117
Processed MSA stems	112
Processed dialect suffixes	6
Processed dialect prefixes	7
Processed dialect stems	62
Distortion of MSA suffixes	6.98%
Distortion of MSA prefixes	5.65%
Distortion of MSA stem	11.72%

When we applied the algorithm on the large sample, we get the following result:

- Sample size = **9915** words.
- Processed MSA suffixes = 923 suffixes.
- Processed Dialect suffixes = 170 suffixes
- Processed MSA prefixes = 1686 prefixes.
- Processed Dialect prefixes =18 prefixes
- Processed MSA stems = 1599 stems.
- Processed Dialect stems = 294 stems
  - Suffixes distortion rate ≈ 15.55%
  - Prefixes distortion rate ≈ 1.06%
  - Stems distortion rate ≈ 2.97%

It is clear that all above result are not accurate. The The reasons are known: Clitics of length one aren't processed, Ambiguous rules aren't applied, all dialect stems aren't presented to the application's *MSA-Dialect stem corpus corpus* yet, and we didn't store all rules of the sample in the application.

6 Discussion

The experiment shows that the distortion of MSA in Sana'ani dialect is not too large. The MSA output is acceptable and our approach works well. MSA in the output represents about 95% of the total words for a small predefined test sample. That is, the not-translated words represent 5% of the sample [percentage 95% and 5% comes from (100-29)/529*100 and 29/529*100 respectively. Where the number 29 refers to the total unconverted words to MSA and 529 is the total words in the input text]. The ambiguous words represent about 32% of the distortion in that sample. Therefore, our approach can do about 68% of translation well. This gives us a simple view about the accuracy that it can be enhanced through different techniques. There are other methods to calculate the efficiency of translation but these numbers clearly shows that the errors are not too small. The reason is that we don't process the ambiguity. These errors teach us that dialect to MSA translation is difficult (but still easier than MSA-to-Dialect translation, Ch1.Ref.8). As any machine translation, ambiguous is well known problem. Here, ambiguous results from mapping one dialect stem and/or rule to two or more equivalents of MSA. For example, Sana'ani dialect word ' لا – lA' it has more than one meaning in MSA. One is 'إلى – <lY ' (means to as in **table 5a**) and the 2nd is as it is in dialect (means no, negation as in **table 5b**). This is a new type of ambiguity. There is no similar in Arabic-to-English machine translation (since there is difference in scripts at least if the pronunciation is equal). We can call the second meaning "unchanged". In addition, there is a 3rd meaning, **table 5c**. We need inference system or machine understanding to decide between such these mapping.

Table 5a Ambiguity of ' لا - *lA*': means *to*

1st meaning of (لا - *lA*) in MSA is *to* " إلى - *<lY* ":					
Sana'ani dialect translit	أخي Axy	عند End	**لا lA**	اسرح AsrH	
MSA	أخي	عند	**إلى**	اذهب	
gloss	brother	round	**to**	go	
Meaning	Go round to my brother				

Table 5b Ambiguity of ' لا - *lA*': means *no,* negation

2nd meaning of (لا - *lA*) in MSA is *not* " لا - *lA*". That is, the same script (no change):						
Sana'ani dialect translit	شي \$y	ولا **wlA**	ابنش Abn\$	مع mE	تتعصبي ttESby	لا **lA**
MSA	شيء	ولا	ابنكِ	مع	تتعصبي	لا
gloss	thing	and **no**	your son	with	gang up	**don't**
Meaning	Do **not** you gang up with your son like that!					

Table 5c Ambiguity of ' لا - *lA*': means *if*

3rd meaning of (لا - *lA*) in MSA is *if* " إذا - <*A ":					
Sana'ani dialect	... اسرح	تكسب ،	تشتي	انت	**لا**
Translit	... AsrH	, tksb	t\$ty	Ant	**lA**
MSA	... اسرح	تكسب،	تشتهي	أنت	**إذا**
Gloss	... go	win	want	you	**if**
meaning	If you want to win, go …				

Second problem is in the approach itself. That is, a rule may need to know neighbors of in-process token and processing the neighbors also may need processing the token first. This we can call it "the dependant rules problem". We should use effective technique to solve this problem. We solve this problem – partially - using the original sentence while processing each token. The other dependant rule will be applicable by removing clitics (stemming) and/or replacing dialect stem with MSA equivalent using corpus. We didn't face complicated dependant rules.

7 Conclusion and Outlook

We have shown how to use rule-based algorithm to translate dialect to MSA. The accuracy is acceptable and it appears to be increased by addressing some weakness in the implementation. This work can be combined with other dialects processing techniques to build an effective tool that is capable of translating dialect to MSA with high accuracy. Rules tags can be expanded to include extra fields in order to remove uncertainty in applying rules. Problem of inputting tags manually can be solved by developing techniques similar to *automatic tagging* used in Arabic [Ch1.Ref.3]. MSA-Dialect stem corpus is still important so large effort should be made to build it and tags should be included in the corpus for each stem.

8 References

[1] K. Taghva, R. Elkhoury, J. Coombs. Arabic Stemming without a Root Dictionary. Information Science Research Institute (ISRI). University of Nevada, Las Vegas.

[2] Tim Buckwalter. 2002. Buckwalter Arabic Analyzer (version 2.0 in 2004). Linguistic Data Consortium (LCD2002L49).

[3] M. Diab, K. Hacioglu, D. Jurafsky. 2004. Automatic Tagging of Arabic Text: From Row text to Base Pharse Chuncks. In 5th Meeting of the North American Chapter of the Association for Computational Lenguistics/Human Language Technologies Conference (HLTNAACL04), Boston, MA.

[4] D. Chiang, Diab, Habash, Rambow, S. Shareef. Parsing Arabic Dialects, 2005. Final Report, 2005 JHU Summer Workshop.

[5] Habash, Rambow. 2005.Tokenization, Morphological Analysis, and Part-Of-Speech Tagging for Arabic in One Fell Swoop. In Proc. of the Association for Computational Linguistic (ACL).

[6] Mohamed. MAAMOURI, Ann BIES. Developing an Arabic Treebank: Methods, Guidelines, Procedures, and Tools. Linguistic Data Consortium (LDC).

[7] Kfir Bar. Machine Translation. May 2006. NLP Seminar.

[8] Mona Diab, Nizar Habash. Arabic Dialect Processing. AMTA 2006, Boston, USA.

[**9**] Abd al-Rahman Mutahhar, Dr. Janet C. E. Watson. 2002. Social Issues in Popular Yemeni Culture. Yemeni-British Project supported by the British Embassy, Social Fund for Development and Leigh Douglas Memorial Fund. Written in modern Sana'ani dialect with English translation.

[10] M. Al-Razi. Mukhtar El-Sehah.1996 .مختار الصحاح. An Arabic-Arabic root dictionary. Al-Resalah publishing House, Beirut, Lebanon.

[11] Charles F Ferguson. 1959. Diglossia. Word, 15(2):325–340.

[12] Nizar Habash, Owen Rambow and George Kiraz. 2005. Morphological Analysis and Generation for Arabic Dialects. Center for Computational Learning Systems. Columbia University.

[13] Shereen Khoja and Roger Garside. Stemming arabic text. Computer Science Department, Lancaster University, Lancaster, UK, http://www.comp.lancs.ac.uk/computing/users/khoja/stemmer.ps, 1999.

[14] Leah Larkey, Lisa Ballesteros, and Margaret Connell. Improving stemming for Arabic information retrieval: Light stemming and co-occurrence analysis. In SIGIR 2002, pages 269t274, 2002.

[15] Leah S. Larkey and Margaret E. Connell. Arabic information retrieval at UMass in TREC-10. In Proceedings of TREC 10, 2002.

[16] William B. Frakes and Ricardo Baeza-Yates, editors. Information Retrieval, Data Structures & Algorithms. Prentice Hall, Englewood CliÌs, NJ, 1992.

An Automatic Punctuation Marks System For Arabic Texts

Hassan Mathkour , M.S. Aksoy, Ameur Touir, and Alaaeldin Hafez
College of Computer and Information Sciences
King Saud University
Riyadh, Saudi Arabia
Mathkour@ksu.edu.sa, aksoy@ksu.edu.sa, touir@ksu.edu.sa, ahafez@ksu.edu.sa

Abstract: This work presents a system for Automatic Arabic punctuation marks. Existing approaches for automatic punctuation marks do not provide suitable performance for and do not satisfy user interests in Arabic texts. The importance and rising need to automate the correct insertion of punctuation marks in Arabic texts led to a need of specific analysis of the Arabic language to introduce approaches that suite the characteristics of the Arabic language. In this paper, we propose an automatic Arabic punctuation mark system based on Arabic text rules. The proposed automated Arabic punctuation mark system is intended for testing, detecting, and placing the correct punctuation marks in the correct place in stripped Arabic sentences. An experiment to evaluate the performance of the system has been conducted using various Arabic texts.

Key words: Arabic punctuation, Lexical cues, Semantic rules

1. Introduction

The rapid growth and wide spread of the concept of computer textual have increased the importance of natural languages processing (NLP). Natural languages processing are used for test and process the language text. This paper presents a new approach to test and detect the Arabic punctuation marks and automatically replace in the correct places in a sentence which stripped of all punctuation. Existing approaches for automatically punctuation marks have been focused in speech recognition [1, 2] with less attractive performance and not satisfactory to user interests in Arabic punctuation marks. To the best of our knowledge, little work has been done on automatic Arabic punctuation marks.

Don Baron, Elizabeth Shriberg, and Andreas Stolcke [1] describe automatic punctuation and disfluency detection in multi-party meetings using prosodic and lexical cues. They have investigated the use of prosodic and word-based classifiers for locating sentence boundaries and disfluencies in multiparty meetings. A lightweight punctuation annotation system for automatic insertion of intra-sentence punctuation into Arabic text is reported in [2]. It demonstrates the feasibility of a system which relies on lexical information. An automatic punctuation method which generates punctuations simultaneously with speech recognition output is presented in [3]. The system produces multiple hypotheses and uses prosodic features to re-score the hypotheses [3]. It is noted that when automatic punctuation is performed with the reference texts, the sequences of word are already given. Therefore, experiments aim at generating punctuation between words.

In this paper, we propose an automatic Arabic punctuation mark system based on Arabic text rules. The proposed automated Arabic punctuation mark system is intended for testing, detecting, and placing the correct punctuation marks in the correct place in stripped Arabic sentences. Arabic text rules are a collection of information about the Arabic text itself. Arabic punctuation marks can be replaced or inserted into stripped Arabic sentences depending on the user's desire based on the Arabic text rules. We demonstrate that, applying Arabic rules for automatically punctuated marks using our methodology achieve a good result compared to existing methods. Current methods of automatically punctuated marks are complex and less accurate. However, using Arabic text rules (as in our method) can solve the problems faced by current methods. We consider the proposed method to be simple yet effective in detecting, and placing the correct punctuation marks in Arabic texts. This is validated by experimenting with various Arabic texts.

2. The Proposed Methodology

Figure 1 shows the steps of our method that automatically tests and detects the correct place of Arabic punctuation marks. The corpuses that were collected and stripped from punctuation marks represent the input of our method. They will be automatically tested and marked by the correct punctuation marks in the correct places by applying the appropriate Arabic rules. Then the results obtained are compared with the original corpuses to validate the working of the system.

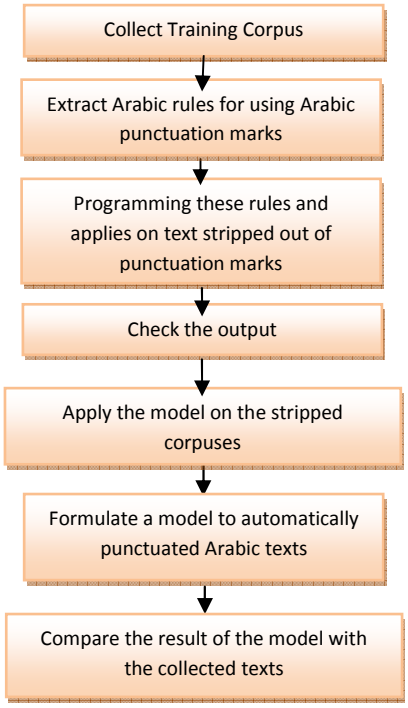


Fig. 1:The process of our methodology

The first step of our method is the extraction of Arabic text rules used for Arabic punctuation marks. Arabic text rules are a collection of information about the Arabic text itself. Arabic punctuation marks can be replaced or inserted into stripped Arabic sentences depending on the user's desire based on the Arabic text rules. Table 1 illustrates the Arabic rules we used in our method and some corpuses. The advantages of using Arabic rules are that to represent the correct way of punctuation marks by applying these rules. Furthermore, Arabic rules require less time to apply with accuracy. In our method we extract Arabic rules by studying many resources such as [4, 5, 6, 7]. Therefore, the results of our method contain information about the Arabic rules. The order of Arabic punctuation marks is given by Arabic text rules, and represented by applying these rules.

The second step of our method is to collect different corpra from different resources such as [8,9,10]. These corpra represented the input for our method after being stripped from punctuation marks. The third step is to recognize and detect the correct place of Arabic punctuation marks by applying the standard Arabic text rules extracted in the first step. The last step of our method is to compare the results of our method (sentences with automatically punctuated marks) with the original corpuses collected.

Table 1: Arabic Text Rules

Arabic Rules	Example
الفاصلة (،) بين الجمل المعطوفة	وكأنهما على موعد مع لقاء أخر على أرض أخرى، وفي معهد من معاهد العلم فيه
النقطة (.) في نهاية الفقرة	خير الناس أنفعهم للناس
الفاصلة المنقوطة (؛) بين جملتين إحداهما سبب حدوث الأخرى	وهو يألم ولكنه لا يشكو ولا يبكي؛ لأنه كان يكره أن يكون كأخته الصغيرة بكّاء شكّاء
النقطتان (:) قبل القول المنقول	قال تعالى : " عليكم أنفسكم "
علامة الاقتباس (" ") يوضع بينهما الكلام المنقول	قال رسول الله صلى الله عليه وسلم: " لا ضرر ولا ضرار "
علامة الاستفهام (؟) بعد صيغة السؤال	لماذا لم تأتي إلى الاجتماع ؟
علامة التعجب (!) للدلالة على معنى متعجب منه	نعم ! إني أعرف طريقي إلى الجامعة

The system has been fully implemented in a Visual Studio 2008.net environment supported by a DBMS to store the corpora. The system has also been experimented with and this is reported below. The program is developed in windows programming using the visual studio.net 2008 Editor with the framework 3.0. The windows programming contains the 3 textbox, in which one textbox contains the corpus and in the second textbox the input and in the third textbox output. The programming technique is to place the rules which are governing to fulfill the Arabic. The technique used is to replace the punctuation marks by finding the related word. There are three strings declared: one string contains the corpus, the second contains the input and the third contains the output of the program. The modification is done according to the input. The method used is the Replace Method which replaces the input with the punctuation mark.

3. Experimenting with the Systems

The database used for our method contains more than 100 different corpora from different resources. For example, different corpora contain several Arabic rules that we apply in our method. Some of the corpora that we use in our method are shown in Table 1. The corpora were stored in database. Experiments demonstrate that Arabic punctuation marks can be automatically detected and replaced in the correct place. An example for applying our method on Arabic text is in Figure 2.

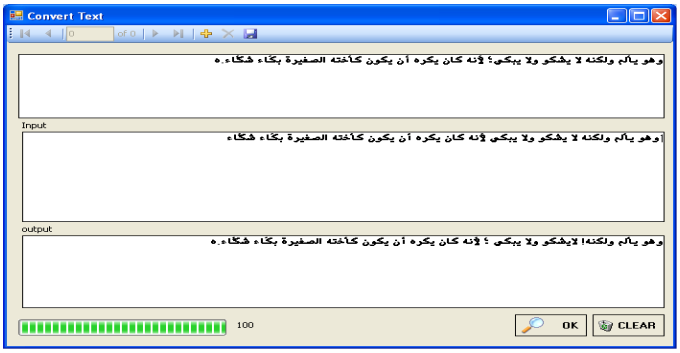


Fig. 2: Automatically Arabic punctuated marks

The first box in Figure 2 belongs to the original corpus, the second box represents the original corpus stripped from punctuation marks (input), and the third box represents the output after applying our

method. The original text (corpus) contains Arabic punctuation marks, the semicolon and Dot (؛ and .), the second box is the input text without Arabic punctuation marks, and in third box we can see the output text after applying our method, which automatically replace the Arabic punctuation marks in the correct place. Figure 3 depicts another example with the original corpus, input Arabic test, and the output after running the system.

Fig. 3: Automatically Arabic punctuated marks

Table 2 shows some of our experimental results which contain some of Arabic test corpora, the stripped text of the punctuation marks as input, and the result as output after applying Arabic rules on the stripped text. The results shown in Table 2 demonstrate the percentage of the correctness of our method when applying Arabic rules on Arabic texts to automatically punctuate the texts.

Table 2: Demonstrate some examples of our experiments

No.	Original corpus	Input	Output	Result (in%)
1	كان آخر الدنيا من هذه الناحية قريبا؛ فقد كانت تنتهي إلى قناة عرفها حين تقدمت به السن، و كان لها في حياته أو قل في خياله تأثير عظيم.	كان آخر الدنيا من هذه الناحية قريبا فقد كانت تنتهي إلى قناة عرفها حين تقدمت به السن و كان لها في حياته أو قل في خياله تأثير عظيم	آخر الدنيا من هذه الناحية قريبا؛ فقد كانت تنتهي إلى قناة عرفها حين تقدمت به السن، و كان لها في حياته أو قل في خياله تأثير عظيم.	100%
2	دَعا أَحَدُ الحُكماءِ أبناءَهُ قُبيلَ مَوتِهِ، وأَخذَ يُحدثُّهمْ بهُدوءٍ. ثُمّ أمرهُمْ أنْ يُحضرّوا حُزمَةٌ من العِيدان، ولمّا إمتثلَ أبناؤُهُ لأمرهِ، طلبَ إلى كُلٍّ منهمْ أنْ يَكسرَ الحُزمة، و لمْ يَقدرْ أحَدٌ على كسرهِ. ففرَّقها وطلبَ إليهمْ أنْ يَكسروها عُوداً عُوداً، و لمْ يَجدوا في ذلِكَ أيّةَ صُعوبَةٍ. عِندئذٍ فقالَ: هَكذا أنتُمْ دائِماً، أقوياءُ إذا اتّحَدتُمْ، ضُعفاءُ إذا تفرّقتم. والدّينُ الإسلامي يأمرنا على الإتّحادِ وعدم التّفرقةِ وعدم التّعصُبِ. قال تعالى (وأعتصموا بحبل الله جميعاً ولا تفرّقوا)، وقال أيضاً (وتعاونوا على البرِّ والتّقوى).	دعا أَحَدُ الحُكماءِ أبناءَهُ قُبيلَ مَوتِهِ و أخذ يُحدثُّهمْ بهُدوءٍ ثُمّ أمرهُمْ أنْ يُحضرّوا حُزمَةٌ من العِيدان و لمّا إمتثلَ أبناؤُهُ لأمرهِ طلبَ إلى كُلٍّ منهمْ أنْ يَكسرَ الحُزمة و لمْ يَقدرْ أحَدٌ على كسرهِ ففرَّقها وطلبَ إليهمْ أنْ يَكسروها عُوداً عُوداً و لمْ يَجدوا في ذلِكَ أيّةَ صُعوبَةٍ عِندئذٍ فقالَ هَكذا أنتُمْ دائِماً أقوياءُ إذا اتّحَدتُمْ ضُعفاءُ إذا تفرّقتم والدّينُ الإسلامي يأمرنا على الإتّحادِ وعدم التّفرقة وعدم التّعصُبِ قال تعالى وأعتصموا بحبل الله جميعاً ولا تفرّقوا وقال أيضاً وتعاونوا على البرِّ والتّقوى	دعا أَحَدُ الحُكماءِ أبناءَهُ قُبيلَ مَوتِهِ و أخذ يُحدثُّهمْ بهُدوءٍ ثُمّ أمرهُمْ أنْ يُحضرّوا حُزمَةٌ من العِيدان و لمّا إمتثلَ أبناؤُهُ لأمرهِ طلبَ إلى كُلٍّ منهمْ أنْ يَكسرَ الحُزمة و لمْ يَقدرْ أحَدٌ على كسرهِ ففرَّقها وطلبَ إليهمْ أنْ يَكسروها عُوداً عُوداً و لمْ يَجدوا في ذلِكَ أيّةَ صُعوبَةٍ عِندئذٍ فقالَ: هَكذا أنتُمْ دائِماً أقوياءُ إذا اتّحَدتُمْ ضُعفاءُ إذا تفرّقتم والدّينُ الإسلامي يأمرنا على الإتّحادِ وعدم التّفرقة وعدم التّعصُبِ. قال تعالى : وأعتصموا بحبل الله جميعاً ولا تفرّقوا وقال أيضاً وتعاونوا على البرِّ والتّقوى	95%
3	وهو يألم ولكنه لا يشكو ولا يبكي ؛ لأنه كان يكره أن يكون كأخته الصغيرة بكاء شكاء.	وهو يألم ولكنه لا يشكو ولا يبكي لأنه كان يكره أن يكون كأخته الصغيرة بكاء شكاء	وهو يألم ولكنه لا يشكو ولا يبكي ؛ لأنه كان يكره أن يكون كأخته الصغيرة بكاء شكاء.	100%

In the second row of Table 2, the different percentage obtained after comparing our results with the original text (corpora) is due to the original text itself. The system detected it and replaced it. In some cases, missing of some punctuation marks in our output has been observed due to the fact that some Arabic punctuation marks have many rules to be applied for each punctuation mark i.e., ambiguity. We

use at least one rule for each mark. Our future work is to resolve the ambiguity among the rules. Figure 4 demonstrates the output after automatic punctuation marks have been performed by the system. As indicated in the figure the accuracy of the system is 93% and above.

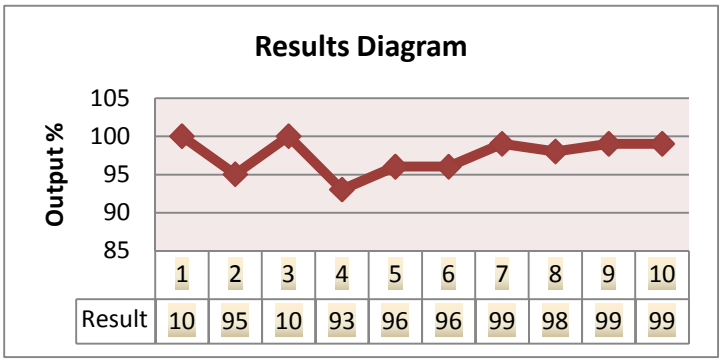


Fig. 4: System Accuracy

4. Conclusion

In this paper, we have proposed an Automatic Arabic punctuation marks system that is based on Arabic text rules. The proposed automated Arabic punctuation mark system is intended for testing, detecting, and placing the correct punctuation marks in the correct place in stripped Arabic sentences.

Compared to existing methods, we have demonstrated that the proposed methodology has achieved good results. The proposed methodology is less complex and more accurate than the current methods, and yet effective in detecting, and placing the correct punctuation marks in Arabic texts. Several experiments have been conducted to evaluate the performance of the proposed methodology compared to current methods, and validate its results with various Arabic texts. The experiments demonstrated that the accuracy of the system is 93% and above.

5. Acknowledgements

The authors wish to thank the research center in the college of computer and information science and the college of computer and information science, King Saud University for partial funding of this work.

6. References

[1]. D. Baron, E. Shriberg, A Stolcke. Automatic Punctuation and Disfluency Detection in Multi-Party Meeting Using Prosodic and Lexical Cues. *International Conference on Spoken Language Processing (ICSLP)*, 2002.

[2]. D. Beerferman, A. Berger, J. Lafferty. CYBERPUNC: A Lightweight Punctuation Annotation System for Speech. *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1998.

[3]. J.H. Kim, P. C. Woodland. The Use of Prosody in Combined System for Punctuation Generation and Speech Recognition. *7th European Conference on Speech Communication and Technology*,2001.

[4]. Punctuation Marks. http://www.saaid.net/mktarat/alalm/3.htm (April. 2009)

[5]. The using of punctuation marks. http://www.almeshkat.net/vb/showthread.php?t=62313 (April. 2009)

[6]. Wikipedia.2009.Panctuation marks. http://ar.wikipedia.org/wiki/%D8%AA%D8%B1%D9%82%D9%8A%D9%85 (April. 2009)

[7]. د. أحمد رضوان,د.عثمان الفريح.التحرير العربي,مطابع جامعة الملك سعود,الرياض 1415هـ

[8]. طه حسين. الأيام. دار المعارف. تونس 1991مـ

[9]. توفيق الحكيم. تحت المصباح الأخضر. دار مصر للطباعة. مصر 1988 مـ

[10]. غالب حمزه أبو الفرج. ذكريات لا تنسى. مطابع الروضة، جدة 1978 مـ

Building an Adaptive Parser for Natural Language Processing:A Message Driven Adaptive Parallel Parsing Engine (MAPPE)

Mohammed El-Affendi
Department of Computer Science
Prince Sultan University
P. O. Box 66833
Riyadh 11586, Saudi Arabia

Abstract- *MAPPE is a top-down parallel parsing engine that has been derived in the context of an experiment on adaptive natural language processing systems. The main idea was to design a parsing engine that copes with dynamic variations in the underlying grammar, avoids backtracking and escapes the overhead of computing complex parse tables. MAPPE satisfies all these requirements. It is independent of the actual form of the input grammar; avoids backtracking by generating all alternatives in parallel and does not require a parse table. The only required table is an internal representation of the input grammar, which may be dynamically modified or replaced without the need to modify the algorithm. Another important feature is that MAPPE is message oriented and conforms well to the recent principles of service oriented architectures (SOA).*

1 INTRODUCTION

One of the major requirements in building an adaptive natural language processing system is that the underlying grammar should be allowed to change and grow dynamically [1]. The implication of this for classical table-driven parsers [2] is that the parse table should be recomputed each time the grammar changes. Accordingly, table driven parsers may not be suitable for this type of system. One algorithm that does not require a parse table is the famous recursive descent parser [2]. Unfortunately this algorithm greatly depends on the actual form of the grammar and is bound to suffer from backtracking in dynamic environments. If these two problems are solved, the algorithm will be suitable for generating adaptive natural language processing systems. MAPPE is an algorithm that rectifies both above problems and transforms the recursive descent process into a message-driven parallel process that is independent of the actual form of the input grammar. The details of the grammar are isolated in a rule base that maintains a direct representation of the production rules and their attributes. Insertions, deletions or modifications of the rule base do not require any changes in the algorithm. This solution is far more suited to dynamic environments than the idea of chart parsers described in [3,4].

MAPPE uses two types of threads to parse a given source: AggregateParseThread and ChoiceParseThread. Normal sequential productions are dealt with using instances of AggregateParseThraed while multiple alternatives for the same nonterminal are dealt with using instances of ChoiceParseThread. The latter forks a child thread for each alternative on the right hand side of the production. The forked child threads compete in parallel to parse the corresponding source segment and a winner is chosen. The progress of the competing parallel threads is synchronized using the position of each in the token stream. Each thread gets the appropriate token depending on its position in the token stream, which is accordingly updated or reset depending on the outcome of the parse attempt.

This paper describes in some detail the implementation of MAPPE and its implication for generating adaptive NLP systems or compilers. The underlying grammar model is described in section 2. An overview of MAPPE is given in section 3. The details of the main coordinating thread are given in section 4. AggregateParseThread and ChoiceParseThread are described in sections 5 and 6 respectively. Synchronization between the competing parallel threads is discussed in section 7. Notes on the implementation of MAPPE are given in section 8. Conclusions, recommendations and implications of the algorithm are given in the final section.

2 RELATED WORK

During the past few years a lot of work has been done in the area of parallel parsing. Most of the reported algorithms either concentrate on the performance issue or the underlying hardware configuration [1], [7]. The conclusion of some researchers is that the improvement in performance may not worth the trouble [13]. Grammar-independence and adaptation are rarely considered. According to [1], parallel parsing approaches may be classified into four broad categories:

2.1 Multiple Serial Parsers:

In this type of parser, a group of serial LR(0) parsers divide an LR(0) stack between them and work in parallel to achieve the parse.

2.2 Process Configuration or Agent Parsers:

In this type of parser a parallel process is assigned to each task in the parser. For example a process may be assigned to each rule in the grammar. The processes then cooperate in parallel to achieve the parse. Most of the current implementations use a bottom-up configuration in which processes try to detect their corresponding rules, and inform their parents or counterparts.

2.3 Connectionist Parsers:

In this type of a parser a trained neural network is used to produce a parsing of a given production rule. A variety of models have been reported in the literature [1], but most of them are used to assist in the parsing process rather than performing the complete work.

2.4 Shared Memory Chart Parsers:

The chart parsing algorithm is well described in [11]. Parallel implementations of the algorithm use a shared-memory that contains the chart and a queue of hypothetical arcs with associated locks. A free processor extracts a hypothetical arc from the queue and checks if it can be conformed, by consulting the chart. If it finds the arc to be genuine, it locks both chart and queue, inserts the arc in the chart, adds some new hypothetical arcs to the queue and unlocks the chart and queue. The process goes on until all arcs are confirmed or an error is detected.

3 THE MAPPE APPROACH

Although MAPPE may be classified as a process configuration parallel algorithm (category 2.2 in section 2 above), it differs from existing algorithms in both objective and approach. The points of differences may be summarized as follows:

1- The major objectives of MAPPE are grammar-independence and backtracking avoidance, although performance improvement is also important.
2- MAPPE is top-down, while most of the reported process configuration algorithms are bottom-up. MAPPE simply maps the parse tree on a similar tree of parallel threads.
3- MAPPE is a full working engine that has been tested and used in other systems, not just an abstract algorithm.
4- MAPPE is message oriented and loosely coupled.

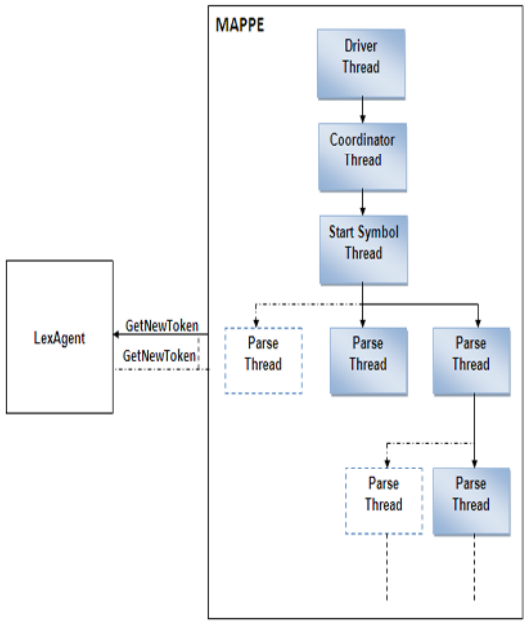


Figure 1: MAPPE Maps the Parse Tree onto an Equivalent Tree of Parallel Threads

4 THE GRAMMAR MODEL

It is assumed that the underlying grammar is an ordinary context-free grammar with the following types of productions (adopting the terminology used in [10]):

5 AN OVERVIEW OF MAPPE

As mentioned above, MAPPE maps the parse tree for a given source onto a similar tree of parallel threads. Figure 1 below shows a typical configuration of the algorithm at run time. The process starts in the Coordinator thread which performs initialization and triggers the StartSymbol thread. The StartSymbol thread starts the parsing process by generating a parse thread for each non-terminal symbol on its right hand side. Two types of parse threads are provided: AggregateParseThread and ChoiceParseThread. AggregateParseThread is a pipeline that is used when the right hand side of a rule is a sequence of symbols. ChoiceParseThread is used when the right hand side of a rule contains several alternatives. ChoiceParseThread deals with all alternatives in parallel with a mechanism to choose a winner. In this way the system avoids backtracking.

MAPPE consists of two major components: a rule-base and a parser. The grammar of the language to be parsed is first stored in the rule base using an object-oriented representation. The parser is then invoked to parse any source or piece of text belonging to the language. Note that the grammar of the language is stored only once

per language and does not change unless the underlying language is changed or modified. The parsing algorithm is independent of the type of language and the actual contents of the rule base.

5.1 The Parser Component: Parser Architecture:

The parser consists of two static threads and a number of dynamic threads that may appear and disappear as implied by the parsing process. The two static threads are initiated when the parser is started and remain running until the parser is terminated. The dynamic threads are created on demand, depending on the underlying grammar.

The static threads are:

1. The **DriverThread** which is mainly responsible for initialization and triggering of the parsing process.
2. The **CoordinatorThread** which is responsible for coordinating and managing the parsing process. It handles parsing requests from individual ParseThreads.

The dynamic threads belong to one of the following classes:

3. **AggregateParseThread** which is created by the CoordinatorThread whenever a request to parse an aggregate production rule is received.
4. **ChoiceParseThread** which is created by the CoordinatorThread whenever a request to parse a choice production rule is received.

All the above threads are implemented as classes. Apart from the DriverThread, all other threads inherit their behavior from a common parent thread class called **ParseThread** the implementation of which is described in the following section.

5.2 The Parent ParserThread:

The parent of all parser threads is defined as follows using Delphi[7] notation:

```
ParseThread = class(TThread)
  protected
    procedure Execute; override;
//Encapsulates the parsing algorithm
  public
    Box: Tlist;  //The message queue
    pname: string;
    BoxEmpty: Boolean;
    function PeekParseMessage:PMessage;
    function GetParseMessage:PMessage;
    procedure PutParseMessage(M:
          PMessage);
    function Empty: Boolean;
    constructor Create(Suspended: Boolean;
       name: string);
    destructor Destroy; Override;
end;
```

Note that the class TThread is a basic class provided by Delphi[7] to encapsulate the behavior of concurrent threads under Windows. The keyword 'override', when used after a function definition, indicates that the function is polymorphic and derived children may override its implementation in the parent class.

As may be anticipated, the methods GetParseMessage and PutParseMessage are used to exchange messages with other threads. GetParseMessage retrieves a message from the thread message queue (Box) and deletes it from the queue. PutParseMessage appends a message to the thread message queue.

The most important thread method is the execute method which is automatically invoked when the thread is created. The major parsing algorithm is encapsulated in this method. As indicated by the override keyword, this method is polymorphic and may be redefined by the inheriting classes.

5.3 Messaging

As indicated in section 3.2 above the parser consists of a collection of concurrent threads that interact to parse a given source. Interaction is achieved through message passing. The types of messages used by the parser threads are summarized in table 1 below. As clear from the table, each message type indicates a specific request or event. Each message in the system is associated with a message structure that further explains the purpose or the task required by the message and provides the parameters required to achieve the task. The message structure is defined as follows:

```
ParseMessage = record
   Sender: TObject;
   mtype: MessageType;
   RuleToParse: Symbol;
   TokenPos: integer;
   Choice: integer;
end;
```

5.5 How Does it Work? A High Level View of the Algorithm

1- The work starts in the DriverThread which performs initialization and creates the CoordinatorThread which is responsible for coordinating the parsing process.

2- The DriverThread then triggers the parsing process by sending a UM_START message to the CoordinatorThread.

3- On receiving UM_START, the CoordinatorThread performs the following:

a- Creates the start symbol thread, which is responsible for parsing the start symbol production rule.

b- Reads in the first input item from the input source.

c- Triggers the start symbol thread by sending it a UM_PARSE message.

d- Wait passively for UM_PARSE, UM_ERROR or UM_SUCCESS messages.

4- On receiving a UM_PARSE message, the start symbol thread and all other descendent threads behave in a similar manner that may be described as follows:

- If the right hand side of the corresponding rule is of the aggregate type (pure or list), ProcessAggregate else if the right hand side is of the choice type then ProcessChoice.
- ProcessAggregate may be summarized as follows:

```
for each symbol on the right hand
side: if the symbol is a nonterminal
then
begin
  create an AggregateParseThread
     and trigger it to parse the
      nonterminal.
  wait for the reply of the child
       AggregateParseThread.
  {The reply may either be
    UM_DONE or UM_ERROR}
end
else if the symbol is a terminal then
ProcessTerminal.
```

- ProcessChoice performs a parallel traverse of all possible parse sub-tree paths and selects a winner. In our implementation the winner is the longest alternative whose corresponding child thread reports success. ProcessChoice may be summarized as follows:

Message	Description
UM_START	Sent by the DriverThread to CoordinatorThread to trigger the parsing process. This message is sent only once in the beginning,
UM_PARSE	Used by a ParseThread to request a parse for a symbol on its RHS.
UM_DONE	Used by a ParseThread to indicate success to the invoking thread.
UM_ERROR	Used by a ParserThread to indicate failure to the invoking thread.
UM_SUCCESS	Sent by the start symbol thread to the CoordinatorThread to indicate that the compilation process terminated successfully.

```
for each alternative on the right
hand side in parallel:
begin
   Create a child
     AggregateParseThread to
      process the alternative.
   Activate the child threads in
      parallel.
   Wait until all child threads
       give their replies.
   Based on the reply of all
   threads decide which reply to
   send to the
   parent:
     UM_DONE or UM_ERROR.
   {The details of how this is
    done is given below}
end
```

Figure 2 below is an object interaction graph [6] for the algorithm. It documents the types and levels of messages that may be exchanged by the objects. The numerical values preceding message names indicate the order of messages in the process. The apostrophe sign ‘ used with some of the level numbers indicates an ‘OR’ situation where only one message of the group (with the same level number) may be sent. A solid box indicates a single object, while a dotted box

indicates a collection of objects. For more on object interaction graphs see [6].

6 WORK OF THE CoordinatorThread

The CoordinatorThread coordinates the parsing process using the following simple algorithm:

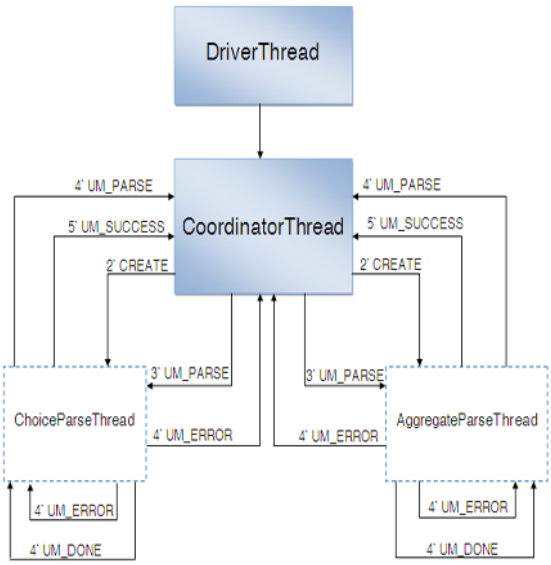


Figure 2: An Object Interaction Graph For MAPPE

```
Algorithm 1: CoordinatorThread execute method:
while not terminated do
    begin
      GetM:=GetParseMessage;
      case GetM^.mtype of
      UM_START:
      begin
        CurrenTSymbol:=GetNextToken; //Read
        the first token of the source
        GlobalSeq:=1; //global token sequence
            number
        Create an AggregateParseThread to process
          the start symbol rule;
          Activate the created thread using a UM_PARSE
message;
     UM_PARSE:
      begin
         RuleToParse:=GetM^.TargetRule;
         case RuleToParse  of
         Aggregate:
              Create an AggregateParseThread to
                process the rule;
          Choice:
               Create a ChoiceParseThread to process
                    the rule;
        end {case};
        Invoke the created thread using a
                  UM_PARSE message;
      end;
    UM_SUCCESS:
      begin
        PrintMessage('Successfuly Parsed');
        Terminate;
      end;
    UM_ERROR:
      begin
        PrintMessage('Errors in Source');
        Terminate;
      end;
    else
        PrintMessage('Unknown Message');
    end; {case}
end; {while}
```

7 THE WORK OF AggregateParseThread:

AggregateParseThread is created by the CoordinatorThread whenever a request to parse an aggregate rule is received from some junior rule thread. It handles both pure aggregates and list aggregates.

The work of this algorithm may generally be described as follows:

- After creation, AggregateParseThread is activated by a UM_PARSE message from CoorddinatorThread or from a parent ChoiceThread.
- On receiving the UM_PARSE message the AggragateParseThread starts processing the RHS of the corresponding rule. It looks at the first symbol on the RHS, and if it is a terminal, it processes it using the ProcessTerminal function, otherwise it processes it using the ProcessNonTerminal function.
- The ProcessTerminal function interacts with the lexical analyzer to process the terminal symbol, and if every thing is alright, AggregateParseThread receives a UM_DONE message from ProcessTerminal. On receiving UM_DONE, AggregateParseThread proceeds to consider the next symbol on its RHS.
- The ProcessNonTerminal function simply sends a UM_PARSE message to the CoordinatorThread on Behalf of the AggregateParseThread. On receiving UM_PARSE, the CoordinatorThread creates a new ParseThread to process the symbol. This new ParseThread may be of type Aggregate or Choice, depending on the type of rule. After processing the symbol, the newly created thread may either return a UM_DONE or UM_ERROR message directly to the requesting AggragateParseThread (not to the CoordinatorThread). On receiving the UM_DONE message, AggragateParseThread proceeds to consider the next symbol on its RHS.
- AggregateParseThread uses a ListInProgress flag to deal with symbols for which the repetition attribute is not ExactlyOnce. This flag is set to

true if the repetition attribute of the current symbol is OneOrMore or ZeroOrMore. When this flag is set and the current symbol is a NonTerminal, AggregateParseThread repeatedly processes the current symbol until a UM_ERROR message is received. A UM_ERROR message in this case indicates the end of symbol replication, and accordingly the ListInProgress flag is reset. As for terminal symbols, the ProcessTerminal function uses the ListInProgress flag to send the appropriate message to the caller and resets the flag when no match is found.

- The process goes on until all symbols on the RHS of the corresponding rule are exhausted.

A more detailed account of the AggregateParseThread parsing algorithm is given in the detailed report.

The polymorphic methods ProcessTerminal and ProcessNonTerminal are implemented as follows for the AggregateParseThread:

7.1 Processing Terminals

The ProcessTerminal function is one of the crucial methods in the work of AggregateParseThread. It interacts with the lexical analyzer through the low level function Admit to process the terminal symbol in question. The token position of the caller is one of the important parameters that should be passed to the lexical analyzer to ensure proper synchronization.
The Admit function returns a positive integer (the position of the next token to match in the token stream) if the processed token matches the current token in the token stream, otherwise it returns a negative integer.
The behavior of ProcessTerminal greatly depends on the repetition attribute of the terminal symbol being processed. As indicated above, there are three possibilities: ExactlyOnce, OneOrMore or ZeroOrMore. Depending on the outcome of Admit and the repetition attribute, ProcessTerminal behaves as follows:

Algorithm 2: ProcessTerminal:

```
procedure ProcessTerminal(S: SymbolType);
 var
   MatchSeq, NoofSymbolsAccepted: integer;
 begin
    NoofSymbolsAccepted:=0;
    repeat
        MatchSeq:=Admit(S,Lseq);
if MatchSeq>0 then {passed token matches current
source token}
begin
   Send a UM_DONE message to the calling thread;
  Update the current token Position: Lseq:= MatchSeq;
      Increment NoofSymbolsAccepted by 1;
    end
    else {No match}
    begin
      if (not ListInProgress) or
        (ListInProgress and (Reptn=OneOrMore) and
(NoofSymbolsAccepted <1))
       then
       begin
         Send a UM_ERROR message to the calling
thread;
       end
      else if ListInProgress then
      begin
        {The mismatch in this case signals the end of
replication, not an error}
        Send a UM_DONE message to the calling thread;
        ListInProgress:=false;
        {Resetting this flag causes ProcessTerminal to
terminate and allows
         the calling thread to proceed with next symbol
on the RHS}
       end;
   end;
  until not ListInProgress;
end; {ProcessTerminal}
```

7.2 Processing NonTerminals:

Processing a nonterminal is much easier than processing a terminal and only requires sending a UM_PARSE message to CoordinatorThread. The procedure may be summarized as follows:

8 WORK OF THE ChoiceThread:

The main difference between AggregateParseThread and ChoiceThread is that the latter deals with multiple alternatives on the RHS for the same construct. It handles production rules of the form:

X → SDF | CGH | PQR

To handle this type of rule, ChoiceThread creates multiple threads, one for each alternative on the RHS. These threads proceed in parallel to achieve the parse. There are three possible outcomes in this case:

1- Only one thread succeeds, which is the most likely result. In this case there will be no problem and the parser will proceed normally.
2- No thread succeeds, which indicates an error situation that may be handled in a normal way.
3- More than one thread succeeds, in which case a decision has to be made as which way to go. In our case the thread that handles the longest alternative (in terms of symbols) is the winner.

9 SYNCHRONIZATION BETWEEN COMPETING PARSETHREADS

One prerequisite for the correctness of MAPPE is that each of the competing ParseThreads gets the appropriate sequence of tokens as if the algorithm were sequential. This is achieved through the condition that each parallel thread maintains a token sequence number that shows its position in the token stream. This token sequence number should be provided to the lexical analyzer whenever the thread is requesting a new token. The lexical analyzer maintains a small buffer and uses a sliding window to fill the buffer with tokens. The synchronization process is built in the GetNewToken function, which is implemented as follows:

```
 function GetNewToken(TokenSeq: integer):
LexSymbolType;
   var
    Ltoken: LexSymbolType;
   begin
     if TokenSeq=GlobalSeq then    //token is not in
buffer
     begin
       Ltoken:=GetNextToken;  // get a new token
       GlobalSeq:=GlobalSeq+1;
       TokenBuffer[GlobalSeq mod BufSize]:=Ltoken;
//add it to buffer
     end
     else
     if (GlobalSeq-TokenSeq)< BufSize then   //token
in buffer
       Ltoken := TokenBuffer[TokenSeq mod BufSize]
//get it
     else
     begin
       MAPPE.Memo1.Lines.Add('LexError');
       Ltoken:=NULToken;
     end;
     Result:=Ltoken;
  end;
```

10 CONCLUSIONS AND RECOMMENDATIONS

A message-driven top-down grammar-independent parallel parsing engine MAPPE) has been designed, implemented and tested. The engine avoids backtracking, does not require a complex parse table and is independent of the actual details of the source grammar. With these properties, MAPPE is ideal for designing adaptive multi-compilers and natural language processing systems. A message-driven approach has been used because messages represent a neat way of passing variable numbers of parameters to procedures and fit nicely within the rising event-driven programming paradigm.

The degree of parallelism is greatly restricted by the sequential nature of the top-down parsing process, but is quite enough to avoid backtracking.

A greater degree of parallelism will be achieved if a bottom-up approach is adopted. An explanation of how this may be achieved is given in [8] which describes BOPP, the bottom-up version of the algorithm.

The complexity of the algorithm is comparable to that of a normal recursive descent parser if we replace function calls in the latter with parse messages.

11 REFERENCES

[1] Adriaens G. & U. Hahn (ed.), Parallel Natural Language Processing, Ablex, Norwood, NJ, 1994.

[2] Aho, A. Sethi, R. & Ullman, J. "Compilers: Principles, Techniques & Tools", Addison Wesley, 1986, USA, pages: 159-277.

[3] Allen, J. "Natural Language Understanding", Second Edition, The Benjamin/Cummings Publishing Company, USA, 1995, pages 54-71.

[4] Borland International, Inc. " The Object Pascal Language Guide", USA 1997.

[5] Calvert, C. "Delphi 2 Unleashed", SAMS Publishing, USA, 1996.

[6] Coleman, D. et al., "Object-Oriented Development: The Fusion Method", Prentice Hall, 1994, pages 63-79.

[7] Dong-Yul Ra and Jong-Hyun Kim. A **parallel parsing** algorithm for arbitrary context-free grammars. *Information Processing Letters*, 58(2):87-96, 22 April 1996.

[8] El-Affendi, M. A. "A Bottom-up Parallel Parser (BOPP) and its Implications", Departmental report MA/Parsing/2, Department of computer science,CCIS,KSU,1997.

[9] El-Affendi, M. A. "Sunbla: An Intermediate Step in a Gradual Promotion Model for Natural Arabic Programming", *AJSE*, Vol 19, No 3,p481-489, KFUPM, Dammam, Saudi Arabia.

[10] Meyer, B. "Introduction to the Theory of Programming Languages", Prentice Hall, USA,1990

[11] Russel, S. & Norvig, P. "Artificial Intelligence: A Modern Approach", Prentice Hall, 1995, pages 696-703.

[12] Tomita, M. "Current Issues in Parsing Technology", Kluwer Academic Publ., Boston, 1991.

[13] Wojciech Rytter. On the complexity of **parallel parsing** of general context-free languages. *Theoretical Computer Science*, 47(3):315-321, 1986.

Improving Classification Accuracy in Random Forest by Using Feature Impurity and Bayesian Probability

Cuong-Nguyen[1], HaNam-Nguyen[2], Wang Yong[1]
[1] College of Economy and Business Administration, Chongqing University, Chongqing , PR China
[2]College of Technology, Vietnam National University, Hanoi, Vietnam

Abstract - *Improved accuracy in data mining tasks is one of the important issues that have been being seized a great attention of many researchers in recent years. The dense forest of the algorithm in data mining generally and data classification specifically, Random Forest seems to be a promising method to implement classification tasks for high-dimensional dataset. In this paper, we use Random, feature impurity and estimation of Bayesian probability as the cardinal elements to build feature ranking formula. After that, we gradually eliminate the feature of lowest position in feature ranking list and compare classification accuracy before and after this elimination. In this way, we build up a best feature subset for the classifier. We conducted the experiments on two public datasets. The results of those experiments show that our proposed method is better than original method as well as some other popular methods, both in classification accuracy and stability.*

Keywords: Data mining, classification, random forest, accuracy, improve

1 Introduction

Improving accuracy in classification tasks is one of interesting topics in data mining. In last several years, the topic has been grasping a lot of attentions from many researchers all over the world. In 2001, Leo Breiman [1] proposed a new algorithm called Random Forest, this algorithm is new approach to data exploration, data analysis, and predictive modeling. This algorithm is combination of three components [2]: (1) CART, (2) Learning ensembles, committees of experts, combining models and (3) Bootstrap Aggregation. Experiments prove that Random Forest's performance is better other previous methods such as: AdaBoost, SVM, Neural Network, C45 [3], etc. Especially, many researchers experimental work [4-5] has proved that Random Forest seems to be very effective in dealing with high-dimensional dataset. The experimental results inspired other researchers, [3, 6-8] tried to improve Random Forest to higher level of classification accuracy as well as to eliminate redundant and noisy features in the classifier and they achieved some remarkable successes.

In this paper, we propose a model by the combination of random forest algorithm, feature impurity (GINI index) and Bayesian probability to improve classification accuracy of the classifier in Random Forest. At the first glance, the method seems akin to method DEF-RF proposed by HaNam-Nguyen et all [3]. Actually, the proposed method is an improvement of DEF-RF and RF to get increase classification accuracy of algorithm, especially in case of imbalance classes which is not in the scope DEF-RF algorithm.

The paper is sectioned as follow: section 2 and section 3 briefly introduce Random Forest algorithm and Bayesian probability, respectively. The proposed method will be presented in section 4, the experimental results will be discussed in section 5. The last section is the conclusion.

2 Random Forest

As mentioned above Random Forest is combination of three components: CART, Learning ensembles, committees of experts, combining models and Bootstrap Aggregation. How does Random Forest deals with classification tasks? To classify a new object from an input vector, put the input vector down each of the CARTs in the forest. Each CART gives a classification, and Random Forest asks the trees "votes" for that class. The forest chooses the classification having the majority votes [1, 9].

In Random Forest each CART is grow as follows:

- If the number of cases in the training set is N, sample N cases at random - but with replacement, from the original data. This sample will be the training set for growing the tree.
- If there are M input variables, a number m<<M is specified in such way that at each node, m variables are selected at random out of the M and the best split is used to split the node. The value of m is held constant during the forest growing. For example if we have a 200 column of predictors, typically we select square root (200), it means we will select only 14 predictors, then we split our node with the best variable among the 23, not the best variable among the 200

- Each tree is grown to the largest possible extent. There is no pruning.

The notable thing here in Random Forest is GINI index, in Random Forest GINI index is used as the splitting criterion and defined as squared probabilities of membership for each target category in the node.

$$\mathrm{GINI}(\mathrm{N}) = \frac{1}{2}\left(1 - \sum_{j} p(w_j)^2\right) \quad (1)$$

Where $p(\omega_j)$ is the relative frequency of class ω_j at node N. It means if all the samples are on the same category, the impurity is zero, otherwise it is positive value. In this paper we will use GINI index as a first key element to build the features ranking formula that will be discussed in chapter 4.

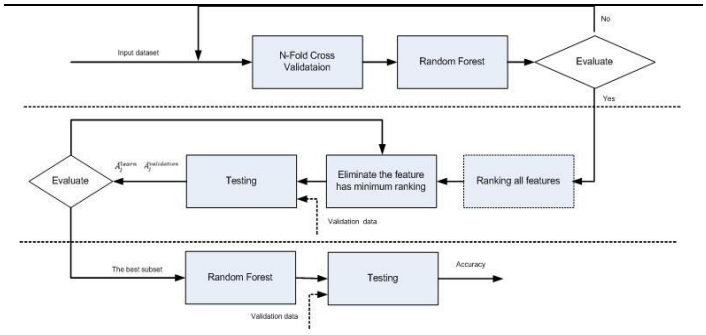
3 Bayesian Probability

Bayesian probability [10] is named after English scientist, Thomas Bayes, who did early work in probability and decision theory during the 18th century. Assume X is an entity and X is described by measurements made on a set of n attributes. H is any hypothesis, such as X belongs to a specified class A. For classification tasks, we want to determine P(H|X), the probability that X belongs to class A, given that we know the attribute description of X. In Bayesian terms, P(H|X) is called the posterior probability of H conditioned on X. In contrast, P(H) is the prior probability of H. The posterior probability, P(H|X), is dependent of X whereas prior probability, P(H), which is independent of X.

Similarly, P(X|H) is the posterior probability of X conditioned on H and P(X) is the prior probability of X. The answerable question is "How can we estimate these probabilities?". Bayes provides an effective method to estimate the probabilities. In practical P(H), P(X|H), and P(X) could be estimated from the given data and the estimation of P(H|X) depends on P(H), P(X|H), and P(X) as follow.

$$\mathrm{P}(\mathrm{H}|\mathrm{X}) = \frac{\mathrm{P}(\mathrm{X}|\mathrm{H})\mathrm{P}(\mathrm{H})}{\mathrm{P}(\mathrm{X})} \quad (2)$$

4 Proposed Method



Our proposed method comprises of four steps akin to DEF_RF algorithm:

1. *Train data by Random Forest with the cross validation*
2. *Calculate the ranking criterion for all features F_i, i=1...n.*
3. *Remove a feature by using Dynamic Feature Elimination function.*
4. *Back to step 1 until reach the desired criteria.*

In step 1, we use Random Forest with n-fold cross validation to train the classifier. In j^{th} cross validation, we obtain a set of (F_j, $A^{learn}_{j,k=1..m}$, $A^{validation}_{j,k=1..m}$). In which, F_j, $A^{learn}_{j,k=1..m}$ and $A^{validation}_{j,k=1..m}$ is feature importance, the learning accuracy of class k^{th} and the validation accuracy of class k^{th} respectively. For example if we need to classify a dataset into 2 classes, using Random Forest with n-fold cross validation in j^{th} cross validation we will obtain a set of (F_j, $A^{learn}_{j,1}$, $A^{learn}_{j,2}$, $A^{validation}_{j,1}$, $A^{validation}_{j,2}$). The classification accuracy of the classifier on class k^{th} is calculated as follow:

$$A_{j,k} = \frac{Number\ of\ features\ of\ j^{th}\ cross\ validation\ classified\ correctly\ on\ class\ k}{Number\ of\ features\ of\ class\ k} \quad (3)$$

In step 2, we will setup a feature ranking formula that is use to rank all features in the dataset. This step is the most important step in our algorithm. It is indispensable to mention that our proposed method uses feature ranking formula as key factor to determine as which feature should be eliminated firstly. In other words, the feature ranking formula will help us in determining which feature may be a noisy/redundancy feature. If a feature has high ranking in the dataset then it will be a useful feature for classifier and otherwise. The weakness of feature ranking formula will lead to the weakness of proposed algorithm because this problem will lead time-consuming of algorithm and other related issues. This problem will be discussed in step 3 in detail.

In reality, a simply method usually will use when we judge whether the feature is useful to the classifier or not (that is classification accuracy of the classifier). The method can best be understood as follow: we add a feature into the classifier and assess classification accuracy of the classifier before and after (add the feature). However, in our situation the question is that how can we have a good estimation of classification accuracy? Especially, in case of high-dimensional dataset, the dataset is classified into many classes and the number of features in each class is very different. In other words, we need to deal with a difficult case in classification tasks then with imbalance classes. In order to deal with this issue, within the scope of this paper we will use Bayesian probability to estimate classification accuracy of the classifier.

Now assume that there are m classes, C_1, C_2, ..., C_m. Given an entity X, X is depicted by m features. According to Bayesian probability, the probability that X belongs to the class C_i is estimated as follow:

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)} \quad (4)$$

P(X) is constant for all classes because we know that the probability of an entity can be classified in to a class are the same, so that only $P(X|C_i)P(C_i)$ need to be estimated. According to Bayes' suggestion in case the prior probabilities of the class are unknown, then it is commonly assumed that prior probabilities of all the classes are equally or in other word we have $P(C_1) = P(C_2) = \ldots = P(C_m)$, and we therefore only need estimate $P(X|C_i)$.

We know that with the given dataset of many attributes, it would be extremely computationally expensive to estimates $P(X|C_i)$. In order to reduce computation in evaluating $P(X|C_i)$, the naive assumption of conditional independence of class is made. This presumes that the values of the attributes are conditionally independent of one another. Thus,

$$P(X|C_i) = \prod_{k=1}^{n} P(X_k|C_i) \quad (5)$$

Or $$P(X|C_i) = P(X_1|C_i).P(X_2|C_i)...P(X_n|C_i) \quad (6)$$

$$P(C_i \mid X) = \prod_{k=1}^{n} P(X_k \mid C_i) = P(X_1|C_i).P(X_2|C_i)...P(X_n|C_i) \quad (7)$$

From (4) and (8) we propose a way to estimate classification accuracy of the classifier on learning set as follow:

$$\overline{A}_j^{learn} = \prod_{k=1}^{n} A_{j,k}^{learn} \quad (8)$$

Similarly, classification accuracy of the classifier on validation set:

$$\overline{A}_j^{validation} = \prod_{k=1}^{n} A_{j,k}^{validation} \quad (9)$$

We propose a new feature ranking formula for feature i^{th} base upon the calculations above

$$F_i^{rank} = \sum_{j=1}^{n} F_{i,j} x \frac{1}{\left(\overline{A}_j^{learn} - \overline{A}_j^{validation}\right)^2 + \varepsilon} \quad (10)$$

Where:

- j=1,.., n is the number of cross validation folders,
- $F_{i,j}$ is GINI index,
- $\overline{A}_j^{learn}$, $\overline{A}_j^{validation}$ is general accuracy of classifier on learning set and validation set,
- ε is the real number with very small value.

The feature ranking formula includes two elements: (1) the first element is GINI index, the element decreases for each feature over all trees in the forest when we train data by Random Forest; (2) the second element is fraction, nominator of the fraction is constant, equals to 1, denominator of the fraction equals $\left(\overline{A}_j^{learn} - \overline{A}_j^{validation}\right)^2 + \varepsilon$, presents the variance between classification accuracy of classifier on learning set and validation set. That means the smaller variance, the better features we have. The combination between GINI index and the fraction presents our expectation: higher ranking feature are better feature. The ε is used to deal the case $\overline{A}_j^{learn}$ equals $\overline{A}_j^{validation}$, in this case $F_i^{rank} = \sum_{j=1}^{n} F_{i,j}$.

After finishing the step 2, we have an ordered list of ranking features. The list will use in step 3 to determine optimal features of the classifier. One should be noted that feature assessing procedure is the correlation among features. We know that a feature may have a low position in feature ranking list but when it is use concurrently with other features they will bring a great contribution to classification accuracy of the classifier. One feasible way to deal with this issue is to use feature elimination strategy which is the next step (step 3) of our proposed method.

In step 3, same as DFE-RF, we also use dynamic feature elimination strategy to eliminate noisy/redundant features. In this step, we will use feature ranking list as a standard criterion to determine which feature should be eliminate first. In other words, the feature of lowest position in feature ranking list will eliminate first. At each step in feature eliminating procedure we will validate the classification accuracy of the classifier. Purpose of the validation is to determine whether the feature to be eliminated is actually redundancy/noisy feature or not. We can perform the validation by comparing the classification accuracy of the classifier before and after eliminating the feature. If classification accuracy of the classifier before eliminating feature is greater than classification accuracy of the classifier after eliminating feature then feature will be kept or otherwise. This iteration will terminate whenever classification accuracy of new subset is higher than classification accuracy of previous subset. Our algorithm will stop when we cannot find out better classification accuracy or no feature to eliminate. In this case the current subset is the best subset we can have. Otherwise, in term of n-fold cross validation the procedure will jump back to step 1 (step 4).

5 Experimental Results

We use R-language[11] as programming language and Random Forest package[12] to validate our proposed method. We validate our proposed method on two public datasets: Medalone dataset, and Colon cancer dataset. In our experiment each dataset was randomly divided into two subsets called learning set and validation set. Random Forest and our proposed method (RF_CT) is executed on both two subset, the achievement we have after execute Random Forest and RF_CT will be used to evaluate classification performance of each method.

5.1 Madelone

MADELON is an artificial dataset containing data points grouped in 32 clusters placed on the vertices of a five dimensional hypercube and randomly labeled +1 or -1 [13]. This dataset is one of five datasets use in the NIPS 2003 feature selection challenge. Actually, Madelon is matrix of 4000 rows x 500 columns that equally a dataset of 4000 instances in which each instance includes 500 attributes.

Table 1: The comparison of some statistical parameters between RF and RF_CT on learning set and validation set of Medalon dataset through 50 testing times with number of trees in RF n=100,150, 200 and 250.

	Mean (%)	**Standard Deviation**	**Min (%)**	**Max (%)**
n=100				
RF	71.24	1.79	66.50	76.50
RF_CT	87.29	0.56	86.00	88.67
n=150				
RF	72.26	1.29	69.67	75.83
RF_CT	87.47	0.05	86.33	88.33
n=200				
RF	72.50	1.32	68.67	74.83
RF_CT	87.70	0.06	86.17	89.00
n=250				
RF	73.11	1.28	70.33	75.83
RF_CT	87.55	0.06	86.00	88.83

Table 1 shows experimental results after 50 testing times with replacements of Random Forest parameter, n, number of trees in Random Forest on Medalon dataset. Generally, we see that proposed method shows better performance than original method both in classification accuracy and stability. In the best case, under framework of our experiment, our method reaches classification accuracy of 87.70±0.06 that seem to be much better than original Random Forest (72.50±1.32).

Figure 1 shows the comparison of classification accuracy between Random Forest and RF_CT on Medalon dataset through 50 testing times and number of trees in Random Forest n=100, 150, 200 and 250 respectively.

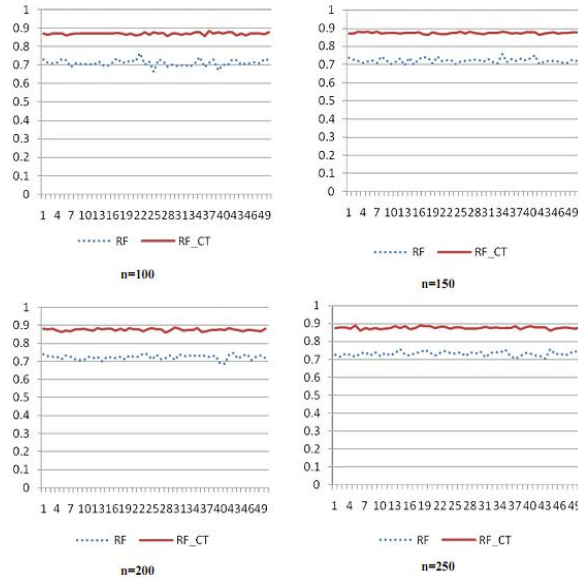


Figure 1: The comparison of classification accuracy between RF and RF_CT on Medalone dataset through 50 testing times and number of trees in RF n=100, 150, 200 and 250.

In last couple of years, some researchers have tested their proposed methods on Madelone dataset [14] and they have achieved some successes but our proposed method seem to be superior on Madelon dataset if the methods are evaluate based on two criterion: classification accuracy and stability.

Table 2: The comparison of classification accuracy among some methods on Madelone dataset

Method	**Classification Accuracy (%)**	**Standard Deviation**
Naïve Bayes	58,3	1,5
C45	69,8	4,7
GOV	71,2	2,9
DOG	71,4	2,6
RF_CT	87,7	0,6

5.2 Colon Cancer

Colon Cancer is also a public dataset [15]. The data set contains 62 samples collected from colon-cancer patients. Among them, 40 tumor biopsies are from tumors (labeled as "negative") and 22 normal (labeled as "positive") biopsies are from healthy parts of the colons of the same patients. Table 3 shows experimental results of Random Forest and RF_CT on the dataset.

Table 3: The comparison of some statistical parameters between RF and RF_CT on learning set and validation set of Colon cancer dataset through 20 testing times with number of trees in RF n=800,1100, 1400 and 1700.

	Mean (%)	Standard Deviation	Min (%)	Max (%)
n=800				
RF	76.17	7.03	56.67	90.00
RF_CT	87.17	5.44	76.67	96.67
n=1100				
RF	78.67	5.76	63.33	86.67
RF_CT	87.17	4.98	76.67	93.33
n=1400				
RF	76.33	8.30	56.67	86.67
RF_CT	86.83	4.52	80.00	93.33
n=1700				
RF	78.17	6.71	66.67	93.33
RF_CT	88.17	3.82	8.00	93.33

Through 20 testing times on Colon cancer dataset our proposed method also presents an impressing achievement in comparison with Random Forest. In case of 1700 of number of trees in Random Forest our method archives the classification accuracy of 88.17±8.82 meanwhile Random Forest is only 78.17±6.71. In other cases that number of trees in Random Forest are 800, 1100 and 1400 our method shows better classification accuracy than Random Forest: 87.17±4.98 and 76.17±7.03, 87.17±4.98 and 78.67±5.76 and 86.83±4.52 and 76.33±8.30. Figure 2 graphs the experimental results of our proposed method on Colon cancer dataset.

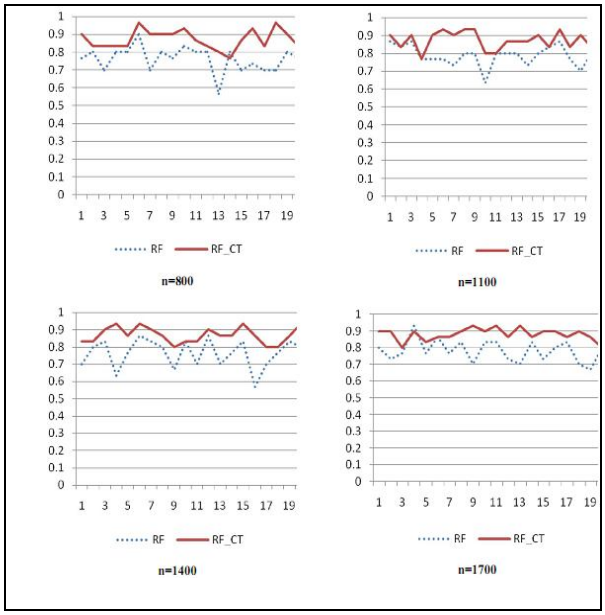


Figure 2: The comparison of classification accuracy between RF and RF_CT on Colon cancer dataset through 20 testing times and number of trees in RF n=800, 11000, 1400 and 1700

Colon cancer dataset is public dataset that widely used as basic dataset to validate new proposed methods in data mining. In fact, many data mining researchers have executed their proposed methods on Conlon cancer data. Table below summarizes some results of some other data mining methods on Colon cancer dataset.

Table 4: The comparison of classification accuracy among some methods on Colon Turmo data

Method	Classification Accuracy (%)	Standard Deviation
GA\SMV	84,7	9,1
Bootstrapped GA\SVM	80	
Combined Kernel for SVM	75,33	7,0
DFE-RF	85,5	4,5
RF_CT	**88,17**	**3,82**

6 Conclusions

Our proposed method presents an improvement of Random Forest and DFE-RF algorithm. In proposed method we took advantages of Bayesian probability and feature impurity to improve classification accuracy in classified tasks. Especially, proposed method also proposes a new approach to improve classification accuracy in case of classification of imbalance classes. Experimental results show our method is better than original method as well as some other popular methods both in classification accuracy and stability.

References:

[1] L. Breiman, "Random Forests," *Machine Learning Journal Paper*, vol. 45, 2001.

[2] X. Su, *Bagging and Random Forests*. [online]. Available: http://pegasus.cc.ucf.edu/~xsu/CLASS/STA5703/notes11.pdf

[3] T. N. V. Ha-Nam Nguyen, S. Y. Ohn,Y. M. Park, M. Y. Han, and C. W. Kim, "Feature Elimination Approach Based on Randon Forest for Cancer Diagnosis," *MICAI 2006: Advances in Artificial Intelligence*, (2006).

[4] W. A. M. S. Kally, "An Optimum Random Forest Model for Prediction of Genetic Susceptibility to Complex Diseases," *Advances in Knowledge Discovery and Data Mining*, vol. 4426/2007, ed: Springer Berlin / Heidelberg, pp. 193-204, (2007).

[5] M. W. A. X. C. Heping Zhang, (2009) Software Willows: a memory efficient tree and forest construction package. *BMC Bioinformatics*. [online]. Available: http://www.biomedcentral.com/content/pdf/1471-2105-10-130.pdf

[6] C. Dahinden, Ed., *An Improved Random Forests Approach with Application to the Performance Prediction Challenge Datasets*.

[7] C. V. Anneleen Van Assche, H. Blockeel,S. S. D. Zeroski, "First order random forests: Learning relational classifiers with complex aggregates," *Machine Learning*, pp.149-182, 2006.

[8] S. G. Isabelle Guyon, Ed., *Feature Selection*. Springer, 2006.

[9] L. Breiman. (2002), *Manual On Setting Up, Using, And Understanding Random Forests V3. {online]*. Available: http://oz.berkeley.edu/users/breiman/Using_random_forests_V3.1.pdf

[10] M. K. Jiawei Han, *Data Mining:Concepts and Techniques*, Second Edition ed.: Diane Cerra, 2006.

[11] *R-Language*. [online]. Available: http://www.r-project.org/

[12] *Random Forest package*. [online]. Available: http://cran.r-project.org/web/packages/randomForest/

[13] *Medalone Dataset*. [online]. Available: http://archive.ics.uci.edu/ml/datasets.html

[14] L. Rokach. (2008)Genetic Algorithm-based Feature Set Partitioning for Classification Problems. *Pattern Recognition*. 1693-1717 . [online]. Available: http://portal.acm.org/citation.cfm?id=1340831

[15] *Colon Cancer Dataset*. [online]. Available: http://www.inf.ed.ac.uk/teaching/courses/dme/html/datasets0405.html

Heuristic for Finding the Tuning Number in Data Mining

Abolfazl Fatholahzadeh
Supelec-Metz, 2 rue Edouard Belin,
F-87070 Metz, France
afzadeh@metz.supelec.fr

***Abstract*—While faster and faster algorithms have been presented for finding frequent patterns, but, the memory issue is almost always neglected. To study this problem, this paper presents our heuristic that maps the input data set onto a smaller one. The essential idea is to determine the right choice of the tuning number by the reward measure introduced in this work. The tuning number helps us to find the maximal number of repeats in the input data set. Experiments done on several large data sets show the benefit of our method in data mining.**

I. Introduction

Mining frequent patterns or itemsets is a fundamental and essential problem in many data mining applications. These applications include the discovery of association rules, strong rules, correlations, sequential rules, episodes, multi-dimensional patterns, and many other important discovery tasks [3].

While faster and faster algorithms have been presented for finding frequent patterns, with the exceptions of the works done in [1], [4], the memory issue is almost always neglected. This paper is about memory issue, however, our work differs from the approaches of [1], [4] for two reasons.

- First, in both mentioned works, the traditional trie data structure are used, Whereas, we employed our own data structure, namely, reduced trie plus (RTP) that outperforms the traditional trie data structure.
- Second, instead of working with the original input data set, we use the smaller data set obtained by the tuning number.

RTP is based on the reduced trie (RT) of Aoe and his colleagues [2], where, it is showed that RT outperforms the traditional trie data structure, form space view. These two reasons let us to say that our work is **new**.

To study the memory issue, this paper presents our heuristic for determining the tuning number that maps the input data set onto a smaller one for being represented into an appropriate data structure. Section 2 presents what we call the tuning number. Section 3 describes our introduced reward function allowing us to determine the tuning number. Using this number for the transformation of the input data set to the smaller one and then using the latter for being stored onto RTP are explained in Section 4. Section 5 summarizes the results of experiments. The paper ends by the conclusions.

II. Tuning number

Given a 2D-text/data set ($\mathcal{T}$) of n variable–length records on an alphabet, in this Section, we introduce what we call the tuning number (h) for mapping $\mathcal{T}$ onto a smaller 2D-text ($\mathcal{T}'$).

A **repeating** substring k in a string x is a substring of x that occurs more than once. A **repeat** in x is a set of repeating substring k of x; it can be specified by the length $p \geq 1$ of k (called **period**) and the locations at which k occurs. Thus in x = `abaabab`, the tuple (2; 1,4) describes the repeat of k = `aba` ($p = 2$) at position 1, 4.

The tuning number helps us to find the value for the period. That is to say, in the context of 2D-text, it helps us to map $\mathcal{T}$ onto another 2D-text $\mathcal{T}'$ under consideration of the following features:

1) The total number of repeats in $\mathcal{T}$ be maximal.
2) Competitive space cost of the data structure representing $\mathcal{T}'$.

In order to explain intuitively the need for the tuning number, let us give a very simple example in the following.

Example 1 (no–good and good tuning numbers): Consider $\mathcal{T}$ given below. Divide each record of $\mathcal{T}$ into $\lceil \frac{m_i}{h} \rceil$ strings, where m_i denotes the length of i^{th} record of $\mathcal{T}$ for $i = 1 \ldots n$, and where each divided-string has at most h symbols.

$$\mathcal{T} = \begin{pmatrix} ACTZ \\ ACT \end{pmatrix} \quad \mathcal{T}'_2 = \begin{pmatrix} AC & TZ \\ AC & T \end{pmatrix} \quad \mathcal{T}'_3 = \begin{pmatrix} ACT & Z \\ ACT & \end{pmatrix}$$

A rapid look at $\mathcal{T}'_2$ and $\mathcal{T}'_3$ tells us that the choice of $h = 3$ is better than $h = 2$. Although, the number of repeats for both above $\mathcal{T}'$ s is 2, but if $\{ACT, Z\}$ is represented by a trie, we obtain (7,6)-graph, whereas its counterpart, namely, $\{AC, TZ, Z\}$ costs a (9, 8)-graph. Recall that the (usual) notation of (a,b)-graph stands for a nodes and b edges (transitions), respectively. Below, we describe how, h can be determined.

III. The reward function

Hereafter, word, string and key are used interchangeably. The set of keys will be denoted by K. $|x|$ denotes the length of string x. $\|K\|$ stands for the total lengths of the keys of K. The set of repeats and non-repeat strings *w.r.t.* h will be denoted by K_h and K'_h.

Example 2 (repeats and no-repeat strings): Considering again $\mathcal{T}$ of example 1, we have: $\mathrm{K}_2 = \{AC\}$, $\mathrm{K}'_2 = \{TZ, Z\}$, $\mathrm{K}_3 = \{ACT\}$ and $\mathrm{K}'_3 = \{Z\}$.

The different values of the repeats *w.r.t.* h (if any), will be represented by

$\Theta_h = \{\theta_{h,i} : i = 1, \ldots, n_h\}$ where $\theta_{h,i} \geq 2$ and $n_h = Card(\mathrm{K}_h)$.

Now the size of $\mathcal{T}'_h$ can be formulated as follows:
$\|\mathcal{T}'_h\| = f(h) + \|K'_h\|$ where $f(h) = \sum_{i=1}^{n_h} |K_{h,i}| \times \theta_{h,i}$.

The weighted sum of the repeats illustrated by $f(h)$, called hereafter the repeat function, indicates a way how the 'best' value for h can determined *i.e.*

$$\textbf{Maximize } f(n) = \sum_{i=1}^{n_h} |\mathrm{k}_{h,i}| \times \theta_{h,i}.$$

We have used the following rule as the heuristic for selecting 'better' h:
Rule: If $h' > h$ and $f(h) > f(h') > 0$, then select h.
This rule is based on the following fact:
Fact 1: If $h' > h$, and $f(h) > f(h') > 0$ then $\|\mathrm{K}'_h\| > \|\mathrm{K}_h\|$.
Proof: Since the three following relations: (1) $\mathcal{T}_h = \mathcal{T}_{h'}$, (2) $f(h) + \|\mathrm{K}_h\| = f(h') + \|\mathrm{K'}_{h'}\|$, and (3) $f(h) > f(h')$ hold, then $\|\mathrm{K'}_{h'}\| > \|\mathrm{K'}_h\|$ holds also.

The algorithm for determining the best h is straightforward. It starts by $h = 2$ and works as follows. Let nf denotes the number of the fields of the current current record (or transaction in data mining parlance) of the input data We write $\max(nf)$ to refer to the maximum number of nfs.
For $h = 2$ to $\max(nf)$ do:

1) For each record of the input data set, first, depending on the value of h and nf, forms a long string as a key. Then store the generated keys into K along with the calculation of the frequency set (Θ).
2) Calculate $f(h)$ the entire records via the formula given above.
3) If $f(h) > f(h+1)$ then returns h, otherwise go to Step1.

Notice that the membership of a key to K is the current implementation (written in the language C) of above Step 1 is assured by the hash-coding data structure, rather than the trie.

The following script, written in GAWK, calculate $f(h)$. The code cuts the input file according to the value h and outputs a file of three fields: (1) key, (2) frequency, and (3) one or more locations (each separated by '|' delimiter) of the key occurring at the string of the input file.

```
#!/usr/bin/gawk -f
BEGIN {OUTF = ARGV[1]h}
function cuth()
{ n = length($0);
  for (i = 1; i <n; i = i+h) {
      x = substr($0,i,h);
      y =  position+i;
      freq[x]++;
      pos[x] = pos[x] y  "|";
  }
  position += n;
}
#-------- Body--------
{cuth()}
END {
  if (outf){
    for (word in freq)
    printf "%s\t%d\t%s\n", word, freq[word],
           pos[word] > OUTF
  }
  for (word in freq) {
    if (freq[word] > 1)
       RepeatWeight += length(word) *
                       freq[word];
    else
       UniqueWeight  +=  length(word);
   }
   printf "h=%d\tRepeatWeight=%d\t
           UniqueWeight=%d\n",
           h,RepeatWeight, UniqueWeight;
  }
```

IV. Using tuning number

Once h is selected, we save the keys of K_h and K'_h as well as the values of Θ_h onto RTP. As mentioned above, the concept of the reduced trie is due to Aoe and his colleagues [2]. RTP is our new data structure. That is to say, in a separate work, non reported here, we have ameliorated Aoe's approach, by `redundancy-removal` technique that reduces the space complexity of the trie-suffixes from `exponential` to `linear`.

A trie is a tree structure in which each transition corresponds to a character of the keys in the presented key set K. A path from the root state to a leaf state corresponds to a key. The trie is formed of nodes and arcs. Arc labels consist of symbols, called characters. An arc labeled c from state s to t is represented by $g(s, c) = t$ defined from $S \times \Sigma$ to $S \cup \{fail\}$, where

- S stands for a finite set of states, represented as a positive number, starting with 1 as value for the initial state.
- Σ stands for the alphabet.
- The absence of transition indicates failure(*fail*).

The notation of the goto function (g) is extended to strings by the conditions.

$$g(s, \varepsilon) = s, g(s, ax) = g(g(s, a), x), \text{ where } a \in \Sigma \cup \{'\#'\};$$
$$x \in (\Sigma \cup \{'\#'\})^\star$$

Where

- The set containing all strings over Σ including zero-length string, called empty string ε is denoted by $\Sigma^\star$.
- '$\#$' ($\notin \Sigma$) is an end-marker symbol which is added to the end of all keys. So no prefix of a key can be a key itself.

Retrieval on a trie advances character by character. This property includes a fast retrieval for variable-length strings and quick unsuccessful search determination.

For a key in K, the state s with $g(s, c) = t$ is a separate state if c is a sufficient character (or arc label) for distinguishing that key from all others in K. So a key $x = yz$ can be divided into two strings y and z by separate node t such that $g(1, y) = t$.

A tree constructed only of the arcs from the root to the separate nodes for all keys in K is called the reduced trie (RT) where the transition function $g(s, c) = t$.

TABLE I
DATABASE CHARACTERISTICS

Database	#Items	Avg. Length	#Records
chess	76	37	3,196
connect	130	43	67,557
mushroom	120	23	8,124
pumsb	7117	50	49,046
pumsb	7117	74	49,046
T10I4D100K	1000	10	100,000
T40I10D100K	1000	40	100,000

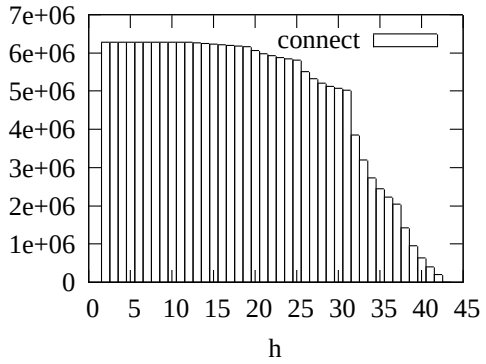


Fig. 1. The evolution of rewards of connect.

The node prior to the separate state is used in our work that allows us to establish a 1-1 relation between the key and information of that key. In our present work, the information of a key (item or itemsets) is the item–counter which is used in finding frequent objects (itemsets, episodes, sequential patterns).

V. EXPERIMENTS AND EVALUATIONS

Experiments were conducted on a divers selection of files (see Table 1) chosen from IBM Almaden [6] (**www.almaden.ibm.com/cs/quest/demos.html**) in which the items(keys) are separated by space-symbols.

For speeding up the time, instead of taking the alphabet of symbols used in these data sets (all are numerical digit), we have considered that the alphabet is composed of the numerical values.

In addition to taking into account the introduced heuristic, instead of taking the alphabet of symbols in used data sets which are the digits, we have considered that the alphabet is composed of the numerical values. That is to say, our alphabet is item-alphabet. This help us to speed up further the time for finding optimal h which in general is long. This is because, as mentioned in Section 3, h starts from 2 to nf: maximum number of the items (or symbols in general term speaking). Figures 1-7 show different values of nf.

Fortunately, for all used data sets, due to our heuristic and to item-alphabet , we rapidly reached to he maximality criteria. That is to say, the value of h with which the reward function is maximized, is 2 selected just after two iterations (h=2 and h=3) as Figures 1-7 show our claim and the usefulness of our heuristic.

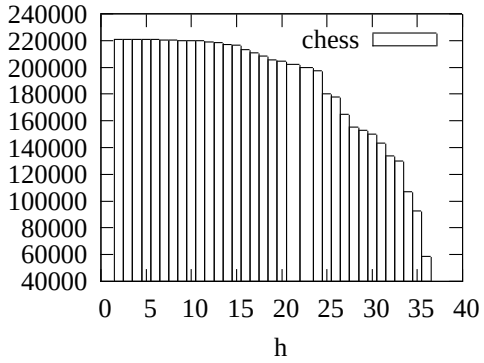


Fig. 2. The evolution of rewards of chess.

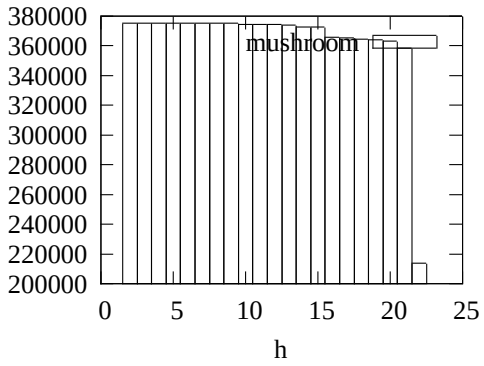


Fig. 3. The evolution of rewards of mushroom.

VI. CONCLUSIONS

This paper proposed a method for mapping an input data set onto a smaller data set by way of the tuning number which can be used in a number of applications, including the data mining. The evaluation of implemented system confirms the benefit of our idea. The proposed heuristic seems to be promising: with few exceptions, we find it useful. The study of a mathematical model for taking into account the exceptions (of the proposed heuristic) is in our agenda.

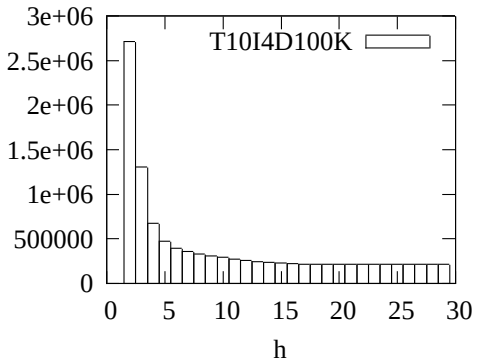


Fig. 4. The evolution of rewards of T10I4D100K.

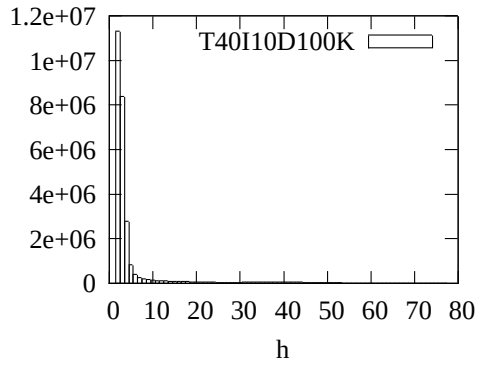


Fig. 5. The evolution of rewards of T40I10D100K.

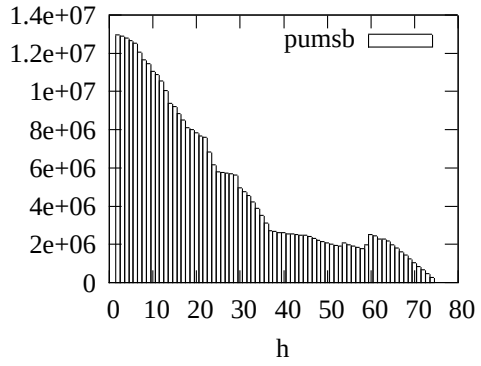


Fig. 6. The evolution of rewards of pumsb.

REFERENCES

[1] A. Amir, R. Feldman and R. Kashi A New and versatile Method for Association Generation.
http://u.cs.biu.il/ Kashi/PAPERS.

[2] J.-I. Aoe, K. Morimotto and T. Sato. An efficient implementation of trie structures. *Software-Practice and Experience*, 22(9):695–721, 1992.

[3] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. Inkeri Verkamo. Fast discovery of association rules. *In U. Fayyad and et al, editors, Advances in Knowledge Discovery and Data Mining*, pages 307328. AAAI Press, Menlo Park, CA, 1996.

[4] F. Bodon and L. Rònyai. Trie: An alternative data structure for data mining algorithms. *Mathematical and Computer Modeling*, 38(7-9):739–751, 2003.

[5] J. Han and M. Kamber. Data Mining: Concepts and Techniques. *Morgan Kaufmann Publishers*, San Francisco, CA, 2001.

[6] Intelligent Information Systems.
www.almaden.ibm.com/cs/quest/demos.html.

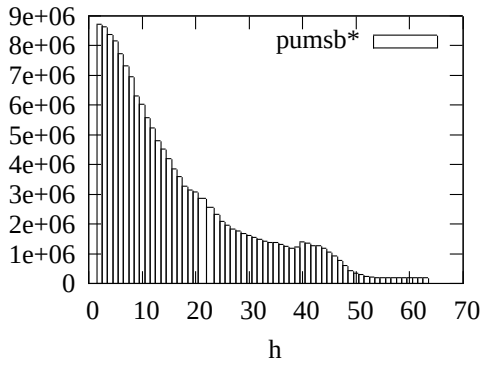


Fig. 7. The evolution of rewards of pumsb-star.

Predicting Breast Cancer Survivability Rates

For data collected from Saudi Arabia Registries

Ghofran Othoum[1] **and Wadee Al-Halabi**[2]
[1]Computer Science, Effat University, Jeddah, Saudi Arabia
[2]Computer Science, Effat University, Jeddah, Saudi Arabia

***Abstract*—** *The application of data mining and machine learning in directing clinical research into possible hidden knowledge is becoming greatly influencial in cancer research. This research presents a comparison of three data mining classification models: multi-layer perceptron neural networks, C4.5 decision trees and Naive Bayes. The classification models are built for breast cancer survivability prediction. The data set used is collected from registries across Saudi Arabia. Due to data scarcity, data synthesis had to be performed using a random seed and a double sampling procedure. After sufficiently preprocessing the data, the classification models were built and three performance measures were used to rank the models: Accuracy, Sensitivity and Specificity. The experiment was set up with multi-layer perceptron as the baseline scheme and with statistical significance of 0.05. Decision Trees performed marginally better than multi-layer perceptron and Naïve Bayes performed significantly worse than the baseline scheme. The result showed that Decsion tree is the the most accurate predictor for breast cancer survivbility in Saudi Arabia (Accuracy 0.979%).*

Keywords: Data Mining, Breast Cancer Prediction, Neural Networks, Decision Trees, Naïve Bayes.

1 INTRODUCTION

With unprecedented growth of data, especially in bioinformatics and the medical field, it has become very important to device computer-based methodologies to analyze meaningful and biologically significant information explosive data banks (Knowledge discovery). Data mining is an algorithmic technique used to describe relationships between patterns and predicting classifications based on the data. Cancer is a malignant disease that is one of the leading causes of deaths worldwide. However, survivability rates in cancer are systematically affected by different interdependent factors. These factors include: genetic factors; genes expressivity which determines phonotypical conditions (including cancer), lifestyle factors and medical history. These factors can be obtained through different data sets.

This research aims at implementing data mining algorithms, applying them to data banks collected from various hospitals in Saudi Arabia, and then evaluating the classification models built. The three prediction models are: neural networks, decision trees and Naïve Bayes.

2 SIMILAR WORK

Till today, there are numerous studies and projects that explore the benefits of using machine learning methods to predict cancer survivability. However, it was noticeable that there are few cancer classification projects completed in Saudi Arabia. The available global researches were investigated in order to formulate a generic structure of how the research can be conducted. Bellaachia and Guven's [1] used statistical data provided by the National Cancer Institute in the United States through SEER (Surveillance Epidemiology and End Results). The data set contained 16 fields in a total of 151,886 records. In their methodology, the authors investigated three data mining algorithms (Naïve Bayes, back-propagated neural network, and C4.5 decision tree algorithm) to predict survivability rates of cancer patients. WEKA (Weikota Environment of Knowledge Analysis) was used to implement the algorithms. K-fold cross validation was used to validate the results. Their hypothesis implies that their results differ from Delen's and Walker's [2]. The hypothesis was verified as the results showed that the methodology that Bellaachia and Guven used outperforms Delen et al approach. Scharber [3] investigated the role of text data mining in predicting cancer survivability. The result of this paper included that using text data mining helps in identifying vital cancer information like cancer type, tumor size and medical history information. These pieces of information, according to Scharber, will improve the treatment plan for each cancer patient. In Zhou,Z & Jiang,Y[4] study, case studies on diabetes, hepatitis, and breast cancer were used in this research paper as the dataset. Neural network ensemble, a

collection of artificial neural networks, was compared to C4.5 Rule-PANE. Neural Network was used as a pre-processor tool and C4.5 Rule-PANE as the main training algorithm. The results showed that C4.5 Rule-PANE is a powerful rule generator. The rules generated by C4.5 have strong generalization ability. Shital Shah, Andrew Kusiak[5] analyzed gene expression data to identify and classify cancer based on its causing genes. For prediction generator, decision tree and support vector machines algorithms were used. The data bank used contained datasets for ovarian, prostate, and lung cancer. This study, integrated all of the algorithms specified in a single gene-finding algorithm. For each type of cancer studied, a set of the most significant gene set was identified with accuracy range of 94-98%.

3 DATA CLEANING AND PREPARATION

Before the dataset is used, it needs to be properly preprocessed and a complete relevancy analysis needs to be completed. Preprocessing entails functions like replacing missing values, normalizing numeric attributes and converting discrete attributes to nominal type. Feature selection involves selecting the attributes that are most relevant to the classification problem. The method used in relevancy analysis is information gain ranker. Below is a detailed presentations of the steps completed in the preprocessing and relevancy analysis (Feature Selection) phases.

3.1 Data Preprocessing

WEKA's filters for feature selection are an integral component in the WEKA package. These filters can either be supervised, belonging to weka.filters.supervised, or unsupervised, belonging to weka.filters.unsupervised. From the unsupervised filters, numeric to nominal conversion was used for nominal attributes and normalization for numerical attributes. Attributes that had more than 70.0% missing values, were omitted. However, some fields' contributions to the pattern were critical and it was considered inefficient to remove them. Table 1 shows the distribution of the class attribute over the collected data from various registries.

1.1 Feature selection

In order to avoid inaccurate or random predictions including redundant, insignificant or noisy attributes, features were selected to be included in the classification (See table 2). This vector of attributes was sent to the Ministry of Health and various hospitals in order to get similar data fields from Saudi Hospital Registries.

Table1 Survivability Distribution in Data Collected from Registries in Saudi Arabia (after random sampling)

Categorical Variable	Frequency	Percentage
Survive Breast Cancer	1260	92.78%
Not survive breast Cancer	98	7.22%
Total	1358	

Table2 Predictor Variables

Attribute	Description
Marital status	This data item identifies the patient's marital status at the time of diagnosis for the reportable tumor.
Birth Place	Patient's birth place
Laterality	The side of the organ in which tumor was detected
Age at diagnosis	
Grade	Differentiation of cells
Radiation	The radiation therapy methodology
Survivability	1 for not survive and 0 for survive
Sex	
Primary Site	The origination site of the tumor

2 PREDICTION MODEL ANALYSIS

2.1 C4.5

J48 in WEKA refers to Quinlan's C4.5 algorithm with optional pruning. J48 is used to build decision trees from a set of labeled training data using the concept of information entropy. To split the data at each stage of the tree construction, a test is performed to select an attribute with the lowest entropy. Information gain (IG) (as shown in equation 3) is used as a measure of entropy (H) with respect to the class attribute (C) :

$$H(C) = -\Sigma p(c) \log p(c) , c \in C \quad (1)$$

$$H(C|Xi) = -\Sigma p(x) [\Sigma p(c|x) \log p(c|x)] \quad (2)$$

$$IGi = H(C) - H(C|Xi) \quad (3) \text{ [1]}$$

In survivability analysis, C would be the survivability class. The choice of the attribute in which the branch is formed depends on a low entropy value and high information gain value. In each iteration, if an entropy value is detected to be higher than in the previous iteration, the tree is pruned. Tree pruning requires the removal of the branch with high entropy value. CS4.5 accurate performance is attributed to its ability to split continuous attributes [7]. Each leaf in a decision tree constructed using C 4.5 is a rule.

2.2 Naïve Bayes

The naive Bayes model is a classical data mining algorithm. It is commonly used to solve prediction problems for its ease of implementation and usage. At the same time, its simplicity doesn't undermine its robustness and effectiveness. Text classification is one of its most common implementations. Throughout the years, it has been through various improvements that are not only reflected in its data

mining capabilities but also in its machine learning pattern recognition.

2.3 Multi-layer perceptron neural network

To solve this non linear classification problem, a multi-layer perceptron with back propagation learning was employed for structuring the model. The network was divided into input, hidden and output layers. In the input layer, the number of neurons was specified by the attributes and the number of output neurons where implied by the possible values of class attributes in this classification. The initial weights were assigned randomly for the connections of the networks and sigmoid function was used as the activation function in order to process the input at each layer and pass it to the next layer using the following equation:

$$\frac{1}{1+e^{-v}} \qquad (2)$$

Where v is the weighted sum of the input nodes.
Five main steps are used in a back propagation- neural network learning phase. These steps are completed iteratively until the error propagated is small enough.

1. Randomly assigning weights to the network
2. Feed forward computation of the activation function
3. Back propagation of the error function to the output layer
4. Back propagation of the error function to the hidden layers
5. Change weights accordingly

3 CLASSIFIER'S EVALUATION

The experimenter module in WEKA shows both text and graphical representation of the results. However, each classifier can have its own additional graphical representations such as decision trees in C4.5. Moreover, the performance parameters that are common amongst all models are: the number of instances that are correctly classified, the number of instances that are incorrectly classified, kappa statistic which measures the agreement of the prediction with the actual classes. Also, there are error measurements such as root mean squared error, mean absolute error, relative absolute error, and root relative squared error.

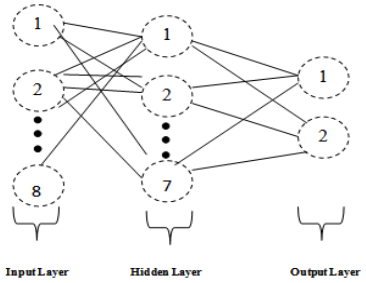


Figure 1 Neural Network Architecture

3.1 Confusion Matrix

confusion matrix is a measure used to tabulate the results of a classifier as true positive, true negative, false positive and false negative. The confusion matrix is built to interpret the results of the classifier. The upper row in a confusion matrix represents the number of instances classified for the positive class and the lower row for the negative class. The true positive cell identifies the attributes that are correctly classified for the positive class where the false positive identifies the classes that are incorrectly classified for the positive class, whilst true negative and false negative represents the attributes that are correctly and incorrectly classified for the negative class respectively. (See table 3)

3.2 KAPPA Statistics

Kappa statistics are used to indicate the correlation between the predicted values and the actual values. These parameters measure the pair-wise agreement between different observed values. [6] Based on the values in the confusion matrix, a kappa value of 1 indicates complete agreement; a kappa value between 0.61 and 0.80 indicates significant agreement. The built classifiers are expected to have a kappa parameter value of more than zero indicating that the predicted classification is not completed by random chance.

3.3 Performance Measures

Three main performance measures are used: accuracy, sensitivity and specificity using the following equations respectively:

1. Accuracy= $\frac{TP+TN}{TP+TN+FP+FN}$ (5)
2. Sensitivity = $\frac{TP}{TP+FN}$ (6)
3. Specificity = $\frac{TN}{TN+FP}$ (7)

Where TP is true positive, TN true negative, FP false positive and FN false negative.
These three measures will be used for this binary classification problem as follows:
Sensitivity will indicate the ratio of how many cases were truly classified as survived out of all which have not been truly classified not survived (true positive and false negative). Specificity will indicate the ratio of how many cases were truly classified as not survived out of those who have not been classified as survived (True negative and false positive). Accuracy will indicate the ratio of truly classified instances out of all instances (true positive, true negative,

false positive and false negative). Sensitivity is referred to as the true positive rate (TPR) and specificity the true negative rate (TNR). Thus, the sum of the TPR and the TNR should equal 1.

The receiver operator characteristics graph is highly used in diagnosis in the medical field as ROC analysis. It compares the quality of different parameters by plotting the TP rate or sensitivity as the independent variable and the FP rate or specificity as the dependant variable. The classifier quality will be assessed based on the area under the graph. As the area under a ROC graph increases for a classifier, the quality of the classifier increases as well.

4 RESULTS

After the three Models were built using 80% split and evaluated using 10 fold cross validation. Decision trees had the highest accuracy and sensitivity with 0.979 and 0.988 respectively. Neural Networks had the highest specificity with 0.896 .The table below shows the detailed performance of the three models.

The ROC Area was lowest for naïve bayes with an area of 0.873(see Figure1) and highest for decision trees with ROC (Receiver Operating Curve) area of 0.952 (see Figure 2). Multi-layer perceptron had a ROC area of 0.9013(see Figure 3).

Table3. Tabular Results of Models' Performance

	Neural Network	Decision Tree	Naïve Bayes
Accuracy	0.978	0.979	0.949
Sensitivity	0.984	0.988	0.950
Specificity	0.896	0.857	0.895
Mean Error	0.0319	0.024	0.094
Kappa Statistic	0.831	0.846	0.479

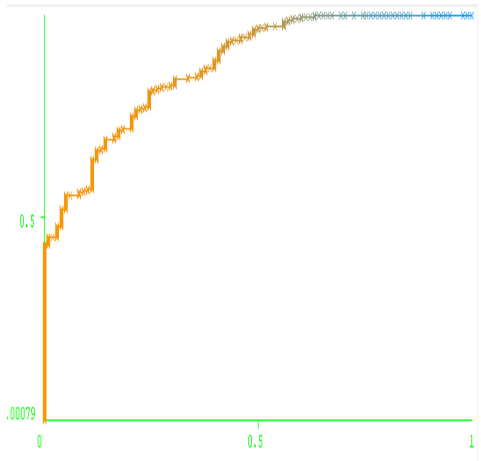


Figure 2. ROC curve for Naïve Bayes after 10-fold cross validation

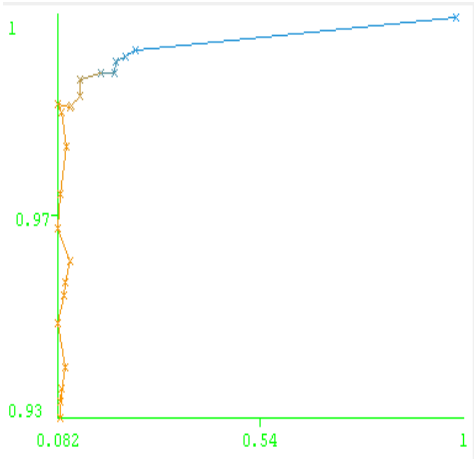


Figure 3. ROC Curve for Decision Tree after 10-fold Cross Validation

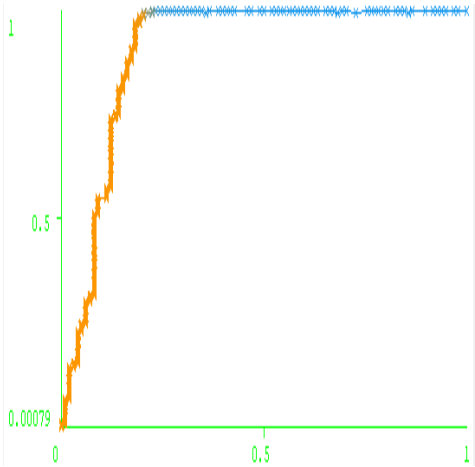


FIGURE 4. ROC CURVE FOR MULTI-LATER PERCEPTRON AFTER 10-FOLD CROSS VALIDATION

7 CONCLUSION

The overwhelming rates of low survivability as a direct result of breast cancer diagnosis is the motivation behind this research especially with a noticed scarcity of the reports and data mining projects complementing the clinical research going in Saudi Arabia.

Results from predictive models are useless without the analysis and feedback of those in the field in order to decide if the results obtained are logical and if they are new findings that are novel in the medical field. Thus, data mining and the medical domain are two integrated areas that complement each other.

Although data mining is becoming a complementary application for many clinical researches in the medical and bioinformatics fields, there are still limitations that can't be ignored. Total dependence on the automation of data mining is not always feasible. Human intervention via interpretation is needed to explore the extracted knowledge. This research can be extended in different ways to increase its usefulness and effectiveness. First, an ensemble of predictive models can be implemented. This will increase the accuracy and will help in introducing novel data mining techniques. Also, the range of survivability could be expanded beyond breast cancer to include survivability rates comparisons amongst different cancer types.

8 REFERENCES

[1] Bellaachia and Guven. 2007. *Predicting cancer survivability using data mining technique*. George Washington University, Department of Computer Science.

[2] Dursun, D. 2005. *Predicting Breast Cancer Survivability: a Comparison of Three Data Mining Methods*. Artificial Intelligence in Medicine, 34 (2), 113-127 .

[3] Scharber, 2007. Evaluation of Open Source Text Mining Tools for Cancer Surveillance: Phase I: Understanding text mining and identifying tools. NPCR-AERRO Technical Development Team.

[4] Zhou, Z & Jiang, Y.(2003). *Medical diagnosis with C4.5 rule preceded by artificial neural network ensemble*. Information Technology in Biomedicine , 7(1), 37 – 42.

[5] Shital Shah &Andrew Kusiak, 2006. *Cancer Gene Search with Data Mining and Genetic Algorithms.* Computers in Biology and Medicine, 37 (2007) 251 – 261.

[6] Jonsdottir,T,et al,2008. *The Feasibility of Constructing a Predictive Outcome Model for Breast Cancer using the Tools of Data Mining.* Expert Systems with Applications, 34(2008),108-118G.

[7] Quinlan, J,R.,1996. *Learning Decision Tree Classifiers.* ACM Comput. Surv. 28(1): 71-72 (1996)

A Feature Selection Method Based on a Support Vector Machine and the Cumulative Distribution Function

Jen-Ing G. Hwang and Chih-En Liu

Department of Computer Science and Information Engineering
Fu Jen Catholic University, Taipei, 24205, Taiwan
E-MAIL: jihwang@csie.fju.edu.tw, lzn97@csie.fju.edu.tw

Abstract:

Feature selection is an important issue in the research areas of machine learning and data mining. It reduces the dimensionality of data and enhances the performance of data analysis and interpretability, such as clustering or classification algorithms. This paper proposes a feature selection method based on support vector machines and distance-based cumulative distribution functions. This method closely relates to the recursive support vector machine (R-SVM) and extends from linear to nonlinear kernels. The proposed method was shown to compete well against R-SVM on both publicly available datasets and high-dimensional proteomic data.

Keywords:

Feature selection; Support vector machine; Cumulative distribution function; R-SVM; Classification

1. Introduction

Feature selection selects a subset of features to reduce data dimensionality and enhance the performance of data analysis and interpretability, such as clustering or classification algorithms. In bioinformatics [1, 3, 5, and 7], the extreme dimensionality of feature space increases the classification uncertainty. For example, the number of samples in genomic or proteomic data is barely comparable with the number of features. Therefore, to accurately manage the high dimensionality of data, feature selection is indispensable.

Zhang et al. (2006) [9] proposed a feature selection method named the recursive support vector machine (R-SVM) to select important genes/biomarkers for biological classification problems. The R-SVM method only works for linear support vector machines (SVMs). This study developed a feature selection method based on SVM [2, 8] using cumulative distribution functions (CDFs) as a distance measure. The modified feature selection method could be applied to linear or nonlinear SVMs.

The distance-based CDF (DCDF) measure is used to determine the discriminatory power of features in binary classification problems. The DCDF method aims to remove irrelevant features in classification problems and increase prediction accuracy. This research implements the DCDF method, applying it to mass spectrometry proteomic datasets and public datasets from the UCI Machine Learning Repository [10].

2. The Method

This section describes the proposed method for feature selection. Firstly, however, the basic concepts of R-SVM are reviewed to provide a foundation for later descriptions of DCDF feature selection.

2.1. R-SVM

SVM is a supervised learning method first introduced for classification purposes and then extended to regression problems. For classification problems, SVMs aim to discover a hyperplane that separates two classes of data samples with the maximum margin possible.

The decision function of a linear SVM may be written as follows:

$$g(\mathbf{x}) = \operatorname{sgn} H(\mathbf{x}) = \operatorname{sgn}\{(\mathbf{W}\cdot\mathbf{x}) + b\} = \operatorname{sgn}\{\textstyle\sum_{i=1}^{n} \alpha_i y_i (\mathbf{x_i}\cdot\mathbf{x}) + b\} \quad (1)$$

where $\mathbf{W}$ and b represent the hyperplane $H(\mathbf{x}) = 0$, $\mathbf{x_i}$ is the input sample i, n is the total number of training samples, $y_i \in \{1, -1\}$ indicates the label of sample i, and $\alpha_i \geq 0$ and b are evaluated from the training samples.

Zhang et al. (2006) [9] proposed the R-SVM method, which is a variant of the SVM recursive feature elimination (SVM-RFE) method [3], designed to select important genes/biomarkers in biological classification problems. To select important features, Zhang [9] defined a score measure to rank each feature based on a linear

SVM, and chose a subset of features with maximum discriminatory power between two sample classes. Consider the hyperplane $H(\mathbf{x}) = 0$ in Eq. (1), the score measure of R-SVM is defined as follows:

$$\mathrm{S} = \frac{1}{n^+}\sum_{\mathbf{x}^+ \in \text{ positive class}} H(\mathbf{x}^+) - \frac{1}{n^-}\sum_{\mathbf{x}^- \in \text{ negative class}} H(\mathbf{x}^-), \quad (2)$$

where n^+ and n^- are the total numbers of positive and negative samples, and $\mathbf{x}^+$ ($\mathbf{x}^-$) denotes a positive (negative) training sample. For the training sample $\mathbf{x}$, the absolute value of $H(\mathbf{x})$ infers a distance from the sample $\mathbf{x}$ to the hyperplane. The score S provides a measure and a larger value of S suggests a better separation between positive and negative samples. From Eq. (1), the value S in Eq. (2) can be calculated as follows:

$$\mathrm{S} = \sum_{j=1}^{p} w_j m_j^+ - \sum_{j=1}^{p} w_j m_j^- = \sum_{j=1}^{p} w_j \left(m_j^+ - m_j^-\right), \quad (3)$$

where m_j^+ and m_j^- are the means of feature j of the positive and negative samples, w_j is the j^{th} component of the weight vector $\mathbf{W}$, and p is the total number of features. Because score S is a weighted sum of $\left(m_j^+ - m_j^-\right)$, the contribution of feature j in S is:

$$S_j = w_j\left(m_j^+ - m_j^-\right), \quad (4)$$

Equation (4) is used to rank features in the R-SVM method. For the detailed description of R-SVM and its procedures, please refer to Zhang [9].

2.2. DCDF Measure

To overcome the limitation of R-SVM only working for linear kernel function, this research develops a SVM-based method using the proposed DCDF measure. The DCDF measure applies the CDF concept to eliminate irrelevant features.

Given a training dataset with positive and negative samples, the SVM finds a trained hyperplane $H(\mathbf{x}) = 0$ that divides the feature space into two parts. On one side of the hyperplane the $H(\mathbf{x})$ sign is positive, while on the other side it is negative. By considering all the positive samples $\mathbf{x}^+$ in the training dataset, this study assumes that the distribution of function values $H(\mathbf{x}^+)$ satisfies the normal distribution model $N(\mu_+, \sigma_+^2)$, where μ_+ and σ_+^2 are the sample mean and sample variance of $H(\mathbf{x}^+)$ for all positive samples $\mathbf{x}^+$. For any sample $\mathbf{x}$ in the training dataset with a function value of $H(\mathbf{x}) = k$, a positive score for k can be defined using:

$$\mathrm{PS}(k) = \mathrm{P}(K^+ \le k), \quad (5)$$

where the random variable K^+ is $N(\mu_+, \sigma_+^2)$, and P is a probability function. Significantly, the definition of PS(k) in Eq. (5) is the CDF.

Similarly, this study assumes that the distribution of function values $H(\mathbf{x}^-)$ satisfies the normal distribution model $N(\mu_-, \sigma_-^2)$, where μ_- and σ_-^2 are the sample mean and sample variance of $H(\mathbf{x}^-)$ for all negative samples $\mathbf{x}^-$. For any sample $\mathbf{x}$ in the training dataset with a function value of $H(\mathbf{x}) = k$, a negative score for k can be defined using:

$$\mathrm{NS}(k) = \mathrm{P}(K^- \ge k), \quad (6)$$

where the random variable K^- is $N(\mu_-, \sigma_-^2)$, and P is again a probability function. Significantly, the definition of NS(k) in Eq. (6) is the complementary CDF.

Equations (5) and (6) are used to evaluate the probability of data belonging to the positive or negative class. If $H(\mathbf{x}_i) = k_i$ for the i^{th} sample $\mathbf{x}_i$, the difference score of $\mathbf{x}_i$ is defined as follows:

$$\mathrm{DS}(\mathbf{x}_i) = \begin{cases} \mathrm{PS}(k_i) - \mathrm{NS}(k_i), & \text{if } \mathbf{x}_i \text{ is a positive sample} \\ \mathrm{NS}(k_i) - \mathrm{PS}(k_i), & \text{if } \mathbf{x}_i \text{ is a negative sample} \end{cases}, \quad (7)$$

Finally, the DCDF measure for a hyperplane $H_{\mathbf{D}}$ trained by dataset D is defined as follows:

$$\mathrm{DCDF}(\mathbf{D}, H_{\mathbf{D}}) = \sum_{i=1}^{n} \mathrm{DS}(\mathbf{x}_i), \quad (8)$$

where n is the total number of training samples. The DCDF measure is used to determine the discriminatory power between the two classes of data. A larger DCDF value indicates a better separation.

2.3. DCDF Feature Selection

By applying the DCDF measure described in section 2.2, the DCDF feature selection eliminates the most irrelevant feature in each recursive step. Before describing the procedure, this study first defines the following notations.

- $\mathbf{D} = \{(\mathbf{x}_i, y_i) | i = 1, \ldots, n, \mathbf{x}_i \in \mathbb{R}^p, y_i \in \{1, -1\}\}$:
  D presents a dataset, y_i is the class label

corresponding to sample $\mathbf{x}_i$, n is the dataset size, and p is the total number of features.

- $\Omega = \{\omega_j | j = 1, \ldots, p\}$:
  Ω presents the feature set of data.
- $H_{\mathbf{D}}$:
  $H_{\mathbf{D}}$ is a hyperplane trained by SVM according to dataset **D**.
- k_i:
  k_i is the function value of $H(\mathbf{x})$ at $\mathbf{x_i}$, also known as $H(\mathbf{x_i}) = k_i$.

DCDF feature selection closely relates to R-SVM and the procedure is described as follows.

Step 1: Input dataset **D** and feature set $\widehat{\Omega} \leftarrow \Omega$.

Step 2: For each feature $\omega_j \in \widehat{\Omega}$

Build the feature set $\widehat{\Omega}_{-j}$ by removing the feature ω_j.

Construct the dataset $\widehat{\mathbf{D}}_{-j}$ with the feature set $\widehat{\Omega}_{-j}$.

Apply SVM on dataset $\widehat{\mathbf{D}}_{-j}$ to find the hyperplane $H_{\widehat{\mathbf{D}}_{-j}}$.

Calculate the $\mathrm{DCDF}_{-j}(\widehat{\mathbf{D}}_{-j}, H_{\widehat{\mathbf{D}}_{-j}})$ score and its error rate.

Step 3: Find the $\widehat{\Omega}_{-j^*}$ that produces the maximum DCDF score at Step 2.

Set $\widehat{\Omega} \leftarrow \widehat{\Omega}_{-j^*}$ and repeat Step 2 and Step 3 until the number of features is 1.

Finally, DCDF reports a subset of features ($\widehat{\Omega}$) according to the lowest error rate in all recursive steps. Although both feature selections of DCDF and R-SVM use recursive strategy, the criteria to choose features are quite different. R-SVM ranks each feature by Eq. (4) and selects relevant features, while DCDF feature selection removes irrelevant features. Moreover, R-SVM only works for linear SVM, while the DCDF measure could apply to SVMs with linear or nonlinear kernel functions.

3. Experiments

In bioinformatics, the data dimensionality is typically significantly larger than the sample size. A feature selection method can select relevant features and solve classification problems. During experiments, this study first compared the performance of DCDF feature selection and R-SVM on real datasets of proteomic data before comparing their performance on public datasets from the UCI Machine Learning Repository [10].

The proteomic data was provided by Dr. Adam at the Center for Biotechnology and Genomic Medicine, Medical College of Georgia, U.S.A. The data was generated by surface-enhanced laser desorption/ionization time-of-flight (SELDI-TOF) mass spectrometry (MS) technology. Each spectrum datum has 60,479 dimensions ranging from 8.58 m/z to 100,301.63 m/z, approximately. Besides healthy normal control (CON), the data also contains values for eight different types of cancer: prostate cancer (PCA), head and neck cancer (SCCA), bladder cancer (BC), colorectal cancer (CC), liver cancer (HCC), lung cancer (LC), pancreatic cancer (PAC), and renal cancer (RC). Due to the nature of MS data, data preprocessing is necessary to eliminate background noise and high dimensionality [4]. This study applied a data preprocessing procedure containing baseline subtraction, normalization, quality assessment [6], peak detection, and peak alignment on spectrum data. During the experiments, eight tests, where each test contrasted one of the eight cancers with the healthy normal control, were performed. Table 1 shows the data preprocessing results for each cancer type versus the healthy normal control. Each preprocessed dataset consists of two groups with 20 individuals per group (40 individuals in total), and the remainder of the dimensionality ranges between 137 and 159.

Table 1. Overview of eight datasets after preprocessing

Test Datasets	Sample size	Features
PCA vs. CON	40	142
SCCA vs. CON	40	147
BC vs. CON	40	154
CC vs. CON	40	141
HCC vs. CON	40	137
LC vs. CON	40	154
PAC vs. CON	40	149
RC vs. CON	40	159

For each test, this study compared the performance of DCDF with R-SVM. Due to the small size of each dataset, every test used the leave-one-out (LOO) cross-validation method. The DCDF used both linear and nonlinear SVM, while R-SVM applied linear SVM only. For nonlinear SVM, this study chose the kernel function as a polynomial of degree 2. The results of the error rates and the number of selected features are shown in Table 2.

For the error rate columns, the cells with a gray background indicate the error rate to be the lowest of each dataset. Similarly, the gray cells in the columns of selected

features indicate the smallest number of selected features of each dataset. Table 2 indicates that the error rates of DCDF with a linear kernel were lower than the rates of R-SVM, except for SCCA. For the tests on CC, HCC, PAC, and RC datasets, DCDF with a linear kernel achieved zero error rates. Regarding the number of selected features, DCDF with a linear kernel selected 5 to 19 features, while R-SVM and DCDF with a polynomial kernel selected a greater amount of features.

Table 2. Experimental results of the proteomic data

Test Datasets	DCDF				R-SVM	
	Linear Kernel		Polynomial Kernel			
	Selected Feature	Error Rate	Selected Feature	Error Rate	Selected Feature	Error Rate
PCA vs. CON	19	0.075	13	0.075	9	0.275
SCCA vs. CON	11	0.1	9	0.1	9	0.05
BC vs. CON	16	0.025	2	0.025	22	0.075
CC vs. CON	8	0	24	0.05	27	0.1
HCC vs. CON	5	0	12	0.05	11	0.075
LC vs. CON	8	0.025	27	0.025	6	0.05
PAC vs. CON	12	0	5	0.05	64	0.05
RC vs. CON	15	0	70	0.075	17	0.125

This study also conducted experiments on four public datasets (Wisconsin Breast Cancer Diagnostic (WBC), Ionosphere (IS), Pima Indians Diabetes (Pima), and BUPA Liver Disorders (BUPA)) from the UCI Machine Learning Repository [10]. Their sample size numbers and features are shown in Table 3.

For the experiments on public datasets, this study applied 10-fold cross-validation to each test; the results are shown in Table 4. The performance of DCDF with a nonlinear kernel outperformed R-SVM. Table 4 indicates DCDF had lower error rates than R-SVM had in WBC, IS, and BUPA datasets, while the error rates on Pima were equal. Regarding the number of selected features, DCDF with a polynomial kernel retained the smallest feature subsets in the WBC, IS, and Pima datasets, while R-SVM selected all features in the IS, Pima, and BUPA datasets.

Table 3. Overview of four datasets from the UCI Machine Learning Repository

Datasets	Sample size	Features
WBC	569	30
IS	351	34
Pima	568	8
BUPA	345	6

Table 4. Experimental results of four UCI datasets

Datasets	DCDF				R-SVM	
	Linear Kernel		Polynomial Kernel			
	Selected Feature	Error Rate	Selected Feature	Error Rate	Selected Feature	Error Rate
WBC	12	0.0246	11	0.0246	24	0.0281
IS	34	0.1054	6	0.0627	34	0.1054
Pima	8	0.2253	5	0.2253	8	0.2253
BUPA	4	0.3072	6	0.2696	6	0.3102

4. Conclusion

To select the important features, this study proposed a DCDF measure and developed a feature selection procedure to retain a subset of relevant features and reduce prediction error rates. The proposed DCDF measure is able to incorporate SVM with either linear or nonlinear kernel functions. The experiment results showed that DCDF outperformed R-SVM under both linear and nonlinear (polynomial) kernel functions. Regarding the experiments using proteomic datasets, DCDF incorporated with a linear SVM performed the best. However, in the experiments on the four UCI datasets, DCDF incorporated with a nonlinear kernel achieved the best performance. The results suggest that when the number of features is larger than the sample size a linear kernel is preferable, otherwise, a nonlinear kernel is recommended.

Acknowledgments

This work was supported in part by the National Science Council, Taiwan, under the grant No. NSC 97-2221-E-030-014. The authors would like to thank Professor Yuan-Chin I. Chang for the helpful discussion and Dr. Bao-Ling Adam for providing the proteomic data.

References

[1] Adam, Bao-Ling, et al. "Serum protein fingerprinting coupled with a pattern-matching algorithm distinguishes prostate cancer from benign prostate hyperplasia and healthy men." Cancer Res, 2002, 62:3609-3614.

[2] Cristianini, Nello, and John Shawe-Taylor. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, 2000.

[3] Guyon, Isabelle, Jason Weston, Stephen Barnhill, M.D, and Vladimir Vapnik. "Gene selection for cancer classification using support vector machines." Machine Learning, 2002, 46:389-422.

[4] Hilario, Melanie, Alexandros Kalousis, Christian Pellegrini, and Markus Muller. "Processing and classification of protein mass spectra." Mass Spectrometry Reviews, 2006, 25:409-449.

[5] Kung, Sun-Yuan, and Man-Wai Mak. "Feature selection for genomic and proteomic data mining." In Machine Learning in Bioinformatics, by Yan-Qing Zhang and Jagath C. Rajapakse. 2008.

[6] Mani, D.R., and Michael Gillette. "Protemoic data analysis: pattern recognition for medical diagnosis and biomarker discovery." In Next generation of data-mining applications, by Mehmed M. Kantardzic and Jozef Zurada. IEEE Press, 2005.

[7] Saeys, Yvan, Iñaki Inza, and Pedro Larrañaga. "A review of feature selection techniques in bioinformatics." Bioinformatics, 2007, 23:2507-2517.

[8] Vapnik, Vladimir. The Nature of Statistical Learning Theory. Springer Verlag, 1995.

[9] Zhang, Xuegong, Xin Lu, and Qian Shi. "Recursive SVM feature selection and sample classification for mass-spectrometry and microarray data." BMC Bioformatics, 2006.

[10] UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/.

In Vitro Fertilization Genetic Algorithm Applied To Multidimensional 0-1 Knapsack Problem

Celso G. Camilo-Junior* **and Keiji Yamanaka†**
* Federal University of Goias, Institute of Informatics, Goiania, GO, Brazil
Email: celsocamilo@gmail.com
† Federal University of Uberlandia, School of Electrical Engineering, Uberlandia, MG, Brazil
Email: keiji@ufu.br

Abstract—In Vitro Fertilization Genetic Algorithm (IVF/GA) is a recently created algorithm that combine the concepts of Genetic Algorithm and In Vitro Fertilization. The IVF/GA has the aim to improve the performance of Genetic Algorithm with the addition of IVF-Inspired mechanisms that better exploit the information of individuals. The Multimensional Knapsack Problem (MKP) is a complex combinatorial problem (NP-hard) and therefore has been addressed in some papers. Thus, this paper applied the IVF/GA, Canonical GA and GA/Busca Tabu to MKP and compared the results. The IVF/GA was the best of them on analyzed instances and therefore we can conclude that IVF/GA is a promising algorithm.

Keywords: In Vitro Fertilization Genetic Algorithm, Knapsack, Genetic Algorithm, Computational Evolutionary.

1 Introduction

Metaheuristics could be classified by the number of potential solutions that are used: solution-based metaheuristics, such as Hill-Climbing, Simulated Annealing, Tabu search; and population-based metaheuristics, such as the Evolutionary Algorithms; or hybrids metaheuristics. All strategies show good results depending on the approached problem.

As an example of metaheuristic, we can cite: Simulated Anneling, Tabu Search, Grasp, VND, VNS and Ant Colony. Among them, Evolutionary Algorithms, especially the Genetic Algorithms, show excellent results and thus they are one of the most popular among researchers.

Genetic algorithms are population-based optimization methods inspired by the mechanisms of alive beings' evolution [1]. The algorithms based on this technique follow the principle of natural selection and survival of the fittest (Charles Darwin).

One of the genetic algorithm advantages is the simplification in the formulation and solution of optimization problems. Simple GAs usually work with coded descriptions formed by bits strings of fixed size. Other types of GAs can work with bits strings of variable size, e.g. GAs used for Genetic Programming [2], [3], [4].

The GA is indicated for the solution of complex optimization problems that involve a large number of variables. However, in some cases, the performance is not satisfactory [5], [6]. Therefore, some works are developed with the aim to improve the performance [7], [8], [9], [10], [11], [12], [13], [14], [15]. Some works focus in the convergence speed and others analyze the efficacy. Furthermore, some papers deal with the preservation of genetic diversity to avoid premature convergence [16], [17], [18], [19].

Considering that In Vitro Fertilization Genetic Algorithm (IVF/GA) is an improved Genetic Algorithm and Multidimensional 0-1 Knapsack Problem (MKP) is a complex problem (NP-HARD), this paper propose the application of the promising IVF/GA [20], [21] to a MKP. Thus, the aim of this paper is to analyze the effects of the IVF-Inspired mechanisms on a complex combinatorial problem.

The paper is organized as follows. One section about the IVF/GA and operators. One section about the MKP. One section about the experiments and results. Finally, the conclusion and suggestions future works.

2 IVF/GA: In Vitro Fertilization Genetic Algorithm

The GAs evolutionary process is composed by a cyclical flow, in which each iteration replaces some individuals with another one. However, many of these eliminated individuals contain genes with information that is important for the search. Therefore, we can say that in every iteration some information is lost - discarded without analysis.

Although this process mimics the evolution of species, it is well-known that there is some loss of information. An aggravating is that the lost information might not go back to the population, causing in some cases a reduction in efficacy and characterizing the loss of opportunity.

To avoid this, the IVF/GA recombines the chromosomes from population of GA and new individuals to better exploit the information presented.

Analogous to IVF the IVF/GA assesses various combinations and selects the individual according its quality. IVF/GA is an algorithm executed in a parallel flow, which receives as input a portion of the GA population and as output it returns an individual, which may be better than the best current (Figure 1). Thus, the IVF/GA could supply the GA of good individuals, improving and accelerating the GA evolutionary process.

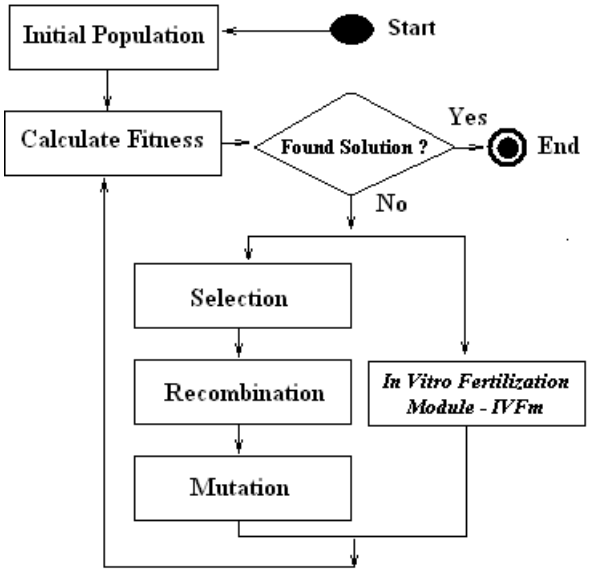


Fig. 1. GA with the IVFm

The ideal way to absorb all the information in individuals would be to recombine them gene-gene, until you find the best combination, which would be computationally unfeasible, especially for larger chromosomes. Therefore, the IVF/GA generate children from the recombination of chromosome parts of some individuals with the best one (section 2.4). If the process generates a better individual, this replaces the current best. Otherwise, there is no interference in the population.

There are two groups of operators for IVF/GA until this moment. The first one with operators that use the generated population by GA as genetic material and the second one with operators that alter chromosomes that will be recombined. The second group strategy is to improve the IVF/GA's population with information that may be beneficial for the recombination process. The operator AR (Assisted Recombination) is a member of the first group and the operators EAR-T, EAR-P and EAR-N are members of the second group.

The IVFm is detailed in the works [20], [21]. Section 2.1 describes the execution flow, section 2.2 describes the division of genetic material, section 2.3 shows the operators and section 2.4 describes the process of recombination.

2.1 The IVFm execution flow

At the beginning of the GA execution, the division of the genetic material is defined (section 2.2) as well as the number of individuals that will be used by IVF/GA (NuIndiv). This configuration is only done in the beginning of the execution of the algorithm.

After receiving the current population of GA, each IVFm generation executes the following tasks:

1) Find the fittest individual and label it as Father;
2) Receive as input the following parameters: the N GA individuals (NuIndiv), which are handled in the recombination process;
3) On the EAR strategy, from N selected individuals, the half (N / 2) changes the chromosomes, forming N'. On the AR strategy N' = N;
4) The fittest (father) and the N' individuals are recombined. As there is only one father for each recombination process, the generated children are siblings;
5) If IVFm generates good individuals, they are inserted in the new population, replacing elected individuals, chosen by the elitism process. Without elitism, individuals replace any others aleatorily.

Task 5 above, explains the interference form of IVFm in the population, generated by GA. The interference only happens when good individuals, better than the best available previously, are generated by the proposed algorithm.

2.2 Division of Genetic Material

Before the recombination (exchange of genetic material) the chromosome is divided into groups of genes or gene to gene. This is called the Genetic Material Division to be exchanged. This is an important step of the algorithm, because we believe that knowledge of the problem helps the designer to choose the best way to divide the chromosome, and so, help the algorithm to solve the problem.

One option is to divide the chromosome according to the coded variables, e.g. if the first 3 genes represent the variable x and the past 2 the y, then the chromosome is divided into two groups, the first one containing 3 genes and the other one 2 (Figure 2). Thus, in recombination process the exchanges are among the variable values (decoded values) and not among encoded values. Therefore, this process provides the designer with a tool for knowledge transferring.

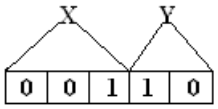


Fig. 2. Two groups selected as the genetic material to exchange

2.3 IVFm's operators

Four independent operators are proposed for the IVF/GA as strategies to help the GA to explore the

search space. The Assisted Recombination Operator (AR) recombines the chromosomes of the current unchanged population. On the Exploratory Assisted Recombination Operators T (EAR-T), all the genes of some chromosomes suffer mutation before recombination process. On the Exploratory Assisted Recombination Operators P (EAR-P), one group of genes, formed by Genetic Material Division, suffer mutation before recombination. Finally, Exploratory Assisted Recombination Operators N (EAR-N) generates new individuals to participate in the recombination process. All EAR operators produce some kind of change in population before the recombination, while the AR operator uses the current population without changes as genetic material. The operators are independent and the combined use of these has not been tested yet.

2.4 The recombination

After establishing the groups of genes (division of genetic material) and the population N', the next step is recombination. In this phase, the chromosome is represented by a vector and every element of this is a group of genes, defined by the division of genetic material. This process is similar to the one applied in the Jung's paper [22].

Analyzing the population N', the best individual is reserved as father and the others as mothers, being the number of mothers limited by the parameter NuIndiv. To create a son, the mother gives an element of its vector and the father supplements the son's chromosome with the other elements of its own vector. This process repeats for all mothers, generating a group of offspring. Other groups are generated by the exchange of the element donated by the mother and the father's chromosome complement, thus, the number of groups of offspring is equal to the number of elements of the father's vector.

After the generation of all groups of offspring, the best child is considered the best Super Individual and it is compared to the father. If better, the son replaces the father in the IVFm's population and the recombination process restarts. If not, the algorithm interrupts the loop and inserts the current father in the GA population.

As an example of a recombination iteration, Figure 3 shows:

- The divided chromosome (division of genetic material) into two parts. Group 1 with 3 genes and group 2 with 2 genes;
- Three mothers established by the parameter NuIndiv = 3, in which the mothers are the three individuals after the father;
- The first group of offspring generated by the donation of the first element of the mother vector and a complement of the father's. For example, the second child from the "Sons of Group 1" (01110) is the result of the union of the first element of the second mother (011) and of the father's complement, i.e. second element of the father vector (10);
- The second group of offspring. In this group, the mother donates the second vector element and the father completes it with its first vector element. For example, the third child from the "Sons of Group 2" (00100) is the result of the union of the second element of the third mother (00) and of the father's complement, i.e. first element of the father's vector (001). In this case, contrary to the "Sons of Group 1", the child is formed by the father's first part and the mother's second part;
- The best individual from the group of offspring is considered the super individual of the group. The best among the super individuals is compared to the father. If it is better than the father, the best super individual replaces the father in the next iteration. If not, the algorithm interrupts the loop and returns the current father.

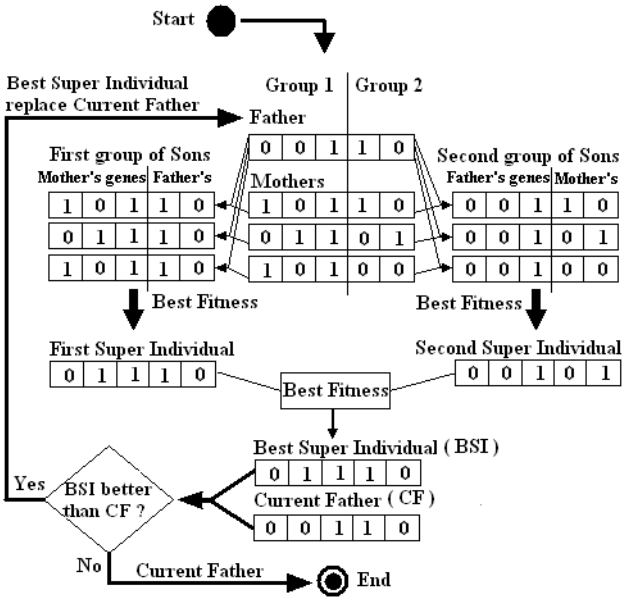


Fig. 3. Example of recombination process

3 Multidimensional Knapsack Problem

The Multidimensional Knapsack Problem (MKP) is a well used benchmark, because of its complexity and similarity with real problems [22].

The NP-hard 01 multidimensional knapsack problem is a generalization of the 0-1 simple knapsack problem [23]. It consists in selecting a subset of given objects (or items) in such a way that the total profit of the selected objects is maximized while a set of knapsack constraints are satisfied. Formally, the problem can be stated as shown on equations (1) to (3).

$$Maximizes \quad \sum_{j=1}^{n} c_j * x_j \tag{1}$$

subject to:

$$\sum_{j=1}^{n} a_{i,j} * x_j \leq b_i, \forall i = 1..m \quad (2)$$

$$x_j \in \{0,1\}, 1 \leq j \leq n \quad (3)$$

where n is the number of objects, m is the number of dimensions that constraints the knapsack, cj represents the benefit of the object j in the knapsack, xj is a binary variable that indicates if the object j has been stored in the knapsack (xj=1) or remains out (xj=0), bi represents the i-th dimension's capacity of the knapsack, and ai,j represents the entries of the knapsack's constraints matrix.

This matrix stores the constraint value for each object in each dimension (weight, size, volume, prize,...). So the matrix has a NxM size: n columns and m rows.

4 EXPERIMENTS

This work use some instances of MKP from OR-Library [24]. For each instance, a scenery is created and ten executions are done. The stop condition is to reach the global optimum or 50000 evaluations (50k).

The chromosome is binary. Each gene represents a object and the function to be maximized is the equation 1 that does not penalize infeasible solutions.

The exclusive use of feasible solutions improves the performance of the algorithm to the problem [25]. To transform the infeasible solutions, a repair operator is created. The operator is based on the work of [26], but less greedy and less deterministic. Instead of always entering the fittest and remove the less fit, our repair operator applies a tournament among individuals to check which will be inserted or removed. The selection pressure of tournament for this problem is 15% for both the exclusion and for insertion.

The initial population is usually an important factor on successful GAs, so we use the same constructive heuristic of repair operator to insert objects in the solution and create the initial individuals. The selection pressure in this case is 20%.

The algorithms Canonical GA (GA), GA/Busca Tabu (BT) and IVF/GA are applied. For a fair comparison, a new initial population is randomly created and shared with the algorithms for each execution. Thus, in each run the algorithms begin with the same initial population. All the executions are independent and sequential.

This work use the Tournament as the selection operator of GA, because it is a method widely used in the literature. For all experiments the PC = 0.8, PM = 1/N and Pop = N*5, where N is the number of objects [25]. BT runs every 10 iterations of GA and the neighbors are chosen randomly. The settings of the experiments are given in Tables 1 and 2.

For better visualization, some acronyms were used. Like QtdG for number of genes, DMG for division of genetic material and QtdIndiv for number of IVF/GA individuals.

4.1 Experimental Results

Some experiments results are collected, like the better value of objective function (MFO), the success rate (QO), the computational time (Ts) and the number of evaluates (Aval). The gap between the best solution of literature and the reached by the algorithms is calculated on second experiment is calculated .

The Table 3 shows the results of algorithms GA, IVF/GA-AR (AR), IVF/GA-EAR-P (EAR-P), IVF/GA-EAR-T (EAR-T) on first experiment. The Table 4 shows the results of algorithms IVF/GA-EAR-N (EAR-N) and GA/BT on first experiment. The Table 5 shows the results of algorithms on second experiment.

Analyzing scenarios 1 and 2 of first experiment, we can identify a similar behavior between the algorithms, they all showed 100% of effectiveness and excellent efficiency, despite the small increase of Ts in the algorithms GA and GA/BT. Analyzing scenario 3, all algorithms have kept the 100% effective and efficient, despite an increase of Ts. We highlight the difference between Ts of GA (0.84 s) and the other algorithms (0.46 s on average).

The scenario 4 remained the effectiveness of 100% of all algorithms, but efficiency varied. We highlight the AR and EAR-N, who spent respectively, 0.13 s and 0.18 s of Ts, moreover 131.4 and 159 of Aval. On Scenario 5, the GA/BT had 90% of effective and the other algorithms 100%. We highlight the performance of EAR-P on Scenario 5, that obtained the lowest Ts and Aval.

On scenario 6 of first experiment, none of the algorithms achieved optimal, probably because it is a problem more complex. Considering that the goal is to analyze the impact of the IVF-mechanisms on GA, the scenario 6 shows that IVF/GA has the best results with highlight for EAR-P with an average of 10,599 Aval(*Gap*% = 0,17%). The GA/BT had the worst performance, probably caused by premature convergence derived from the addition of local search.

On scenario 7 four algorithms achieved the optimum at least once. The EAR-P with 20% effective, AR with 10%, the EAR-N with 20% and the GA/BT with 10%. Analyzing the average of MFO, IVF/GA had the best results: EAR-P was the best (*Gap* = 0.12%), followed by the EAR-N (*Gap* = 0.16%), AR (*Gap* = 0.22%) and EAR-T (*Gap* = 0.26%). The GA/BT was better than the GA, but was worse than the IVF/GA.

On experiment 2, the EAR-P was the best for instances with $\alpha = 0.50$, however, the AR was the best in most of the other. Therefore, AR had the best average of the experiment. On instances with highest rate ($\alpha = 0.75$), the AR had the best performance among the scenarios.

The results of the GA presented in this work were better than presented in the work of [27]. We believed that the configuration assigned to the GA in this work favored the performance. The settings used in this work were based on [25].

TABLE 1
Configuration of the algorithms for the first experiment

Experiment 1									
Scenarios	GA Parameters		BT Parameters		IVF/GA Parameters		Problem Parameters		
	Pop	QtdG	BTMax	Tabu	DMG	QtdIndiv	N	M	Best
1	5*N	N	100	10	2	20%Pop	6	10	3800
2	5*N	N	100	10	2	20%Pop	10	10	8706.1
3	5*N	N	100	10	2	20%Pop	10	15	4015
4	5*N	N	100	10	2	20%Pop	10	20	6120
5	5*N	N	100	10	2	20%Pop	10	28	12400
6	5*N	N	100	10	2	20%Pop	5	39	10618
7	5*N	N	100	10	2	20%Pop	5	50	16537

TABLE 2
Configuration of the algorithms for the second experiment

Experiment 2									
Scenarios	Problem Parameters			GA Parameters		BT Parameters		IVF/GA Parameters	
	N	M	α	Pop	QtdG	BTMax	Tabu	DMG	QtdIndiv
2.1 a 2.10	5	100	0.25	50	N	50	2	2	20%Pop
2.11 e 2.12	5	100	0.50	50	N	50	2	2	20%Pop
2.13 e 2.14	5	100	0.75	50	N	50	2	2	20%Pop

TABLE 3
The results of GA, EAR-T, EAR-P and AR on first experiment

		First Experiment											
		GA			EAR-T			EAR-P			AR		
Scenario		**MFO**	**Ts**	**Aval**	**MFO**	**Ts**	**Aval**	**MFO**	**Ts**	**Aval**	**MFO**	**Ts**	**Aval**
1	**Av**	3800	0.05	42	3800	0.05	47	**3800**	**0.05**	**47**	3800	0.05	51
	QO	10/10			10/10			**10/10**			10/10		
2	**Av**	8706.1	0.32	325	8706.1	0.26	326.8	**8706.1**	**0.15**	**199**	8706.1	0.20	248.4
	QO	10/10			10/10			**10/10**			10/10		
3	**Av**	4015	0.84	735	4015	0.57	614.3	**4015**	**0.46**	**511.3**	4015	0.44	442
	QO	10/10			10/10			**10/10**			10/10		
4	**Av**	6120	0.50	400	6120	1.69	1570.4	**6120**	**0.97**	**894.2**	6120	0.13	131.4
	QO	10/10			10/10			**10/10**			10/10		
5	**Av**	12400	14.86	10402	12400	7.05	5524.8	**12400**	**3.05**	**2583.6**	12400	3.81	3169.8
	QO	10/10			10/10			**10/10**			10/10		
6	**Av**	10589.6	59.05	50115	10592.6	49.98	50111.8	**10599**	**51.77**	**50128.3**	10596	54.83	50133.6
	QO	0/10			0/10			**0/10**			0/10		
7	**Av**	16473.4	66.17	50000	16494	61.41	50159.8	**16516.4**	**58.39**	**48520.2**	16500.6	60.83	49205.4
	QO	0/10			0/10			**2/10**			1/10		

5 Conclusion

The IVF/GA is a genetic algorithm that combines the IVF-inspired mechanisms with cycle evaluation of evolutionary algorithms. The power of IVF was mapped to IVF/GA as a parallel flow that improves the use of genetic information.

Thus, this paper applied the IVF/GA to a complex problem, the Multidimensional 0-1 Knapsack Problem, to test the ability of the algorithm on a combinatorial problem.

Analyzing all tests done, we can observe that the IVF/GA is the promising genetic algorithm. Thus, has a great potential to improve different kinds of binary-coded evolutionary algorithms. Therefore, we suggested the joining of the IVF/GA with different evolutionary algorithms for different types of applications as future work.

References

[1] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional, January 1989.

[2] E. RODRIGUES, "Genetic programming in the optimization of digital circuits (in portuguese)," in *IV Encontro Nacional de Inteligncia Artificial*, Campinas, Brazil, 2003.

[3] C. FERREIRA, "Gene expression programming: A new adaptive algorithm for solving problems," *Complex Systems*, vol. 13, no. 2, pp. 87–129, 2001.

[4] ——, "Gene expression programming in problem solving," Universidade dos Aores, Portugal, Tutorial, 2001. [Online]. Available: http://www.propesq.ufpe.br/anais/anais.htm

[5] M. GEN and R. CHENG, *Genetic Algorithms and Engineering Design*. EDA, 1997.

[6] Z. MICHALEWICZ, *Genetic algorithms + data structures = evolution programs*. Berlin: Springer-Verlag, 1996.

[7] J.-B. PARK, J. PARK, Y.M. abd WON, and K. LEE, "An improved genetic algorithm for generation expansion planning," *Power Systems*, vol. 15, no. 1, pp. 916–922, 2000.

[8] W. RONG-LONG and O. KOZO, "Solving facility layout problem using an improved genetic algorithm," *Fundamentals of Electronics, Communications and Computer Sciences*, vol. E88-A, no. 2, pp. 606–610, 2005.

[9] C. RAJAN, M. MOHAN, and K. MANIVANNAN, "Improved genetic algorithm solution to unit commitment problem," in *Transmission and Distribution Conference and Exhibition*. Asia Pacific: IEEE/PES, 2002, pp. 255–260.

[10] A. S. WU, H. YU, S. JIN, K. C. LIN, and G. SCHIAVONE, "An incremental genetic algorithm approach to multiprocessor scheduling," *Parallel and Distributed Systems*, vol. 15, no. 9, pp. 824–834, 2004.

TABLE 4
The results of EAR-N and GA/BT on first experiment

		First Experiment (continuation)					
		EAR-N			GA/BT		
Scenario		MFO	Ts	Aval	MFO	Ts	Aval
1	Av	3800	0.05	51	3800	0.05	42
	QO	10/10			10/10		
2	Av	8706.1	0.17	214.4	8706.1	0.29	325.1
	QO	10/10			10/10		
3	Av	4015	0.40	392.4	4015	0.44	452.6
	QO	10/10			10/10		
4	Av	6120	0.18	159	6120	1.01	1170.5
	QO	10/10			10/10		
5	Av	12400	8.46	6173.6	12399	10.92	13031.5
	QO	10/10			9/10		
6	Av	10595.7	68.38	50165	10585.6	38.56	50125.9
	QO	0/10			0/10		
7	Av	16510.9	77.07	47055.2	16492.8	42.57	50985.2
	QO	2/10			1/10		

TABLE 5
The results of the algorithms on second experiment

	Second Experiment											
	GA		EAR-T		EAR-P		AR		EAR-N		GA/BT	
Scen	*Gap%*	Ts	*Gap%*	Ts	*Gap%*	Ts	*Gap%*	Ts	*Gap%*	Ts	*Gap%*	Ts
2.1	0.558	189.969	0.656	241.36	0.213	216.657	**0.455**	**193.015**	0.394	233.625	1.046	44.625
2.2	0.101	158.437	0.202	228.674	0.091	207.51	**0.202**	**180.86**	0.101	230.32	0.610	67.83
2.3	0.234	190.94	0.242	240.82	0.301	215.22	**0.119**	**194.66**	0.119	234.54	0.119	69.17
2.4	0.954	192.54	0.988	245.26	1.126	219.795	**0.699**	**194.965**	1.037	231.665	1.557	67.28
2.5	0.277	191.395	0.288	241.735	0.217	216.505	**0.217**	**194.475**	0.369	236.385	0.329	67.56
2.6	0.154	192.16	0.049	246.46	0.154	220.05	**0.049**	**195.69**	0.049	237.17	0.154	69.21
2.7	0.584	194.305	0.334	243.36	0.477	220.02	**0.557**	**198.515**	0.539	242.17	0.807	69.065
2.8	0.000	193.375	0.171	244.31	0.000	219.055	**0.000**	**195.79**	0.000	238.49	0.092	68.945
2.9	0.516	193.5	0.570	243.39	0.743	219.62	**0.516**	**195.33**	0.273	238.06	0.834	68.56
2.10	0.283	192.78	0.553	240.84	0.283	217.84	**0.000**	**193.43**	0.283	234.64	0.291	69.53
Av	**0.366**		**0.405**		**0.361**		**0.281**		**0.316**		**0.584**	
2.11	0.122	193.835	0.122	191.225	0.122	197.91	**0.122**	**205.605**	0.122	280.085	0.122	71.545
2.12	0.235	187.35	0.221	180.13	0.209	183.65	**0.235**	**186.14**	0.235	256.85	0.613	68.22
Av	**0.178**		**0.171**		**0.165**		**0.178**		**0.178**		**0.368**	
2.13	0.000	171.6	0.000	163.65	0.000	178.81	**0.000**	**200.35**	0.000	325.34	0.301	72.05
2.14	0.269	185.22	0.272	162.44	0.351	183.12	**0.179**	**197.97**	0.280	330.38	0.288	72.35
Av	**0.135**		**0.136**		**0.176**		**0.089**		**0.140**		**0.295**	
Av	**0.306**		**0.333**		**0.306**		**0.239**		**0.271**		**0.512**	

[11] Z. RUTTKAY, A. EIBEN, and P. RAUE, "Improving the performances of gas on a ga-hard csp," in *Workshop on Studying and Solving Really Hard Problems*, Cassis, Frana, 1995, pp. 157–171.

[12] J. YEN, J. C. LIAO, B. LEE, and D. RANDOLPH, "A hybrid approach to modeling metabolic systems using a genetic algorithm and simplex method," *IEEE Trans. on Syst., Man, and Cybern.*, vol. 28, no. 2, pp. 173–191, 1998.

[13] R. YANG and I. DOUGLAS, "Simple genetic algorithm with local tuning: Eficient global optimizing technique," *J. Optim. Theory Appl.*, vol. 98, no. 2, pp. 449–465, 1998.

[14] M. MUSIL, M. J. WILMUT, and R. CHAPMAN, "A hybrid simplex genetic algorithm for estimating geoacoustic parameters using matched-field inversion," *IEEE Journal of Oceanic Engineering*, vol. 24, no. 3, pp. 358–369, 1999.

[15] W. T. CHAINATE and P. PONGCHAROEN, "A new heuristic for improving the performance of genetic algorithm," *Academy of Science, Engineering and Technology*, vol. 21, no. 1, pp. 217–220, 2007.

[16] W. A. TACKETT and A. CARMI, "The unique implications of brood selection for genetic programming," in *IEEE World Congress on Computational Intelligence*, Orlando, FL, USA, 1994, pp. 160–165.

[17] S. MAHFOUD, "Crowding and preselection revisited," in *Parallel Problem Solving from Nature 2*, R. Manner and B. Manderick, Eds., Amsterdan, 1992, pp. 27–36.

[18] H. SHIMODAIRA, "An empirical performance comparison of niching methods for genetic algorithms," *IEICE Trans Inf Syst*, vol. E85-D, no. 11, pp. 1872–1880, 2002.

[19] S. MAHFOUD, "Niching methods for genetic algorithms," Department of General Engineering, University of Illinois at Urbana-Champaign, Tech Report, 1995.

[20] C. G. CAMILO and K. YAMANAKA, *Evolutionary Algorithms*. INTECH, 2011, ch. In Vitro Fertilization Genetic Algorithm.

[21] ——, "Assisted recombination: Accelerating genetic improvements of populations in a genetic algorithm," in *Proceedings of the 10th IASTED International Conference on Intelligent Systems and Control*. IASTED, 2007.

[22] S. H. Jung, "Queen-bee evolution for genetic algorithms," *Electronics Letters*, vol. 39, no. 6, pp. 575–576, March 2003.

[23] M. R. Garey and D. S. Johnson, *Computers and Intractability : A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences)*. W. H. Freeman, 1979.

[24] C. C. Petersen, "Computational experience with variants of the balas algorithm applied to the slection of r&d projects," *Management Science*, vol. 13, pp. 736–750, 1967.

[25] A. Hoff, A. Lkketangen, and I. Mittet, "Genetic algorithms for 0/1 multidimensional knapsack problems," in *Proceedings Norsk Informatikk Konferanse, NIK '96*, 1996.

[26] P. C. Chu and J. E. Beasley, "A genetic algorithm for the multidimensional knapsack problem," *Journal of Heuristics*, vol. 4, no. 1, pp. 63–86, June 1998.

[27] S. Khuri, T. Back, and J. Heitkotter, "The zero/one multiple knapsack problem and genetic algorithms," in *Proceedings of the 1994 ACM Symposium on Applied Computing*. ACM Press, 1994, pp. 188–193.

Development of a Diagnostic Expert System for Autism Disorder-PCADEX

Sadaf Sajjad, Hira Qamar, Khadija Tariq, Saira Bano

Department of Humanities
COMSATS Institute of Information Technology, Islamabad, PAKISTAN
sadafsajjad@comsats.edu.pk,

Abstract

Autism is a neurodevelopment, psychological and behavioral disorder. It is a childhood disorder and diagnosing autism is a major issue as its symptoms can easily be confused with those of mental retardness. Successful intervention requires correct diagnosis. As the major issue faced in Pakistan is of proper diagnosis of the autistic children who are mostly misdiagnosed as mentally retarded and thus are unable to get the required treatment which can enable those children to live like a normal child. Despite autism being a major area of concern worldwide, no remarkable work is done in Pakistan as yet. The diagnoses of autism currently being done in Pakistan is based on manual procedures. It is the need of time to have automated systems for diagnosing these disorders instead of doing them manually. Therefore, we developed an expert system, on the requirement of the clinical psychologist, which can act intelligently and correctly diagnoses the subject on the basis of the characteristics/symptoms. It is an interview based system which works on the basis of the behavioral characteristics of autism that are incorporated into the system in the form of rules and facts. Rules and facts make the knowledge base of the system from which the decision is extracted by the inference engine depending on the user's data input. The result is generated on its basis that categorizes the subject as having autism or not and classifies the level of autism present. This expert system will facilitate the clinicians and the related professionals in diagnosing this disorder. And will also open new horizons for the further developments for autism in Pakistan.

1. Introduction

Autism is defined with different terminologies such as a neurodevelopment disorder, neuropsychological disorder and behavioral disorder. Neurodevelopment disorder means that there is developmental deficit in those brain regions which are responsible for the normal child behaviors. Thus the individual with autism has behavioral, social and communication impairments. [1] Neuropsychological and behavioral disorder terminologies are also often used to describe autism in which there is impairment in the socio-cognitive skills and hence abnormal behaviors are observed in these affected individuals. [1]

Autism is diagnosed based on the presence of certain behaviors & communication level either directly observed or more typically reported during parental interview. [7] There is no biological test to diagnose autism, depending on the history and neurological examination; tests are prescribed to the affected persons [1].

Many of the screening and diagnostic tools have been made for the diagnosis of autism but most of these tools are able to detect the syndrome by the age of 18 months and a reliable detection can be made around three years of age [22]. And no instruments can reliably detect the syndrome in first 6-12 months of life because all of these instruments mainly focus on the behavioral attributes of autism which are not shown until age one year.

The major tests/tools or systems that are being used worldwide are:

Autism Diagnostic Observation System (ADOS) and Autism Diagnostic Interview (ADI) are the most comprehensive diagnostic systems for diagnosis. These systems take detailed interview from the individuals and provide observation methods to objectively assess the social, communication and behavioral abilities of the individuals [1].

Checklist for Autism in Toddlers (CHAT) is a brief checklist with good discriminating capacity screening for autism and this tool was very useful as it gave low rate of false positive results [22].

Childhood Autism Rating Scale (CARS) is the best validated instrument for diagnosis of ASD's (Autism spectrum disorders). It is based on behavioral interview and observations [22].

Autism Treatment Evaluation Checklist (ATEC) is an online form which diagnoses autism and also evaluates the treatments progress.

There are a wide range of diagnostic tools developed using artificial neural network (ANN) techniques for the identification of autism. These ANNs consists of synapses which are assigned weights, and of neurons. They are used for information processing after they have learned

and have been trained from typical examples and thus helping in diagnostic purposes e.g.

Self-Organizing Maps (SOMs) are artificial neural networks that are used to develop cortical feature maps [23].

Non-linear pattern recognition system based on neural network technology is used for the classification of autism.

Developmental behavior check list neural network (DBC NN) is a diagnostic system which assesses behavioral and emotional problems in children and adults with a developmental disorder from age 2 to 18 years.

A neurofuzzy based model is also used for the identification or diagnosis of autism.

An expert system called as Knowledge Base Screener is also developed for the early identification of autism in children. The screener is simple and effective method of analyzing child's development and it also help parents in early identification of any developmental disorder [28]. This system was specifically made to target the Indian masses and its marketing is limited to India, not available internationally for general use.

Artificial intelligence (AI) techniques have been usefully employed in bioinformatics and also provided solution to many biological problems. One of the largest areas of applications of artificial intelligence is in Expert Systems also known as knowledge based systems. Expert system makes extensive use of specialized knowledge to solve problem at the level of human expert. Expert system consists of two main parts: the knowledge base and the inference engine. The knowledge base contains all the knowledge on the basis of which inference engines draws conclusion. It contains heuristic knowledge and factual knowledge. Factual knowledge is the knowledge extracted from books, journals and magazines whereas heuristic knowledge is based on one's judgments. Some inferences (facts) are also made by expert system in the same way that a human expert would reason the solution to a problem.

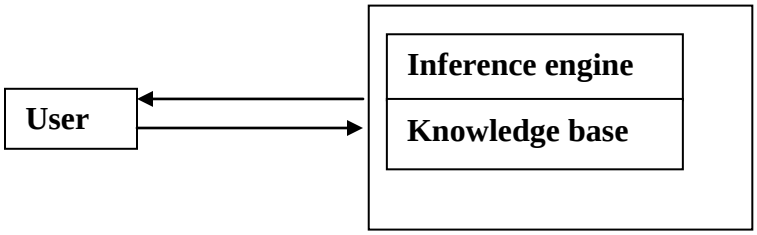


Fig 1.2: Basic function of expert system

The user supplies the desire information to the system and the expert systems acts intelligently and gives response to the user. The knowledge base is designed such that it acts like an intelligent assistant to the human expert. The more knowledge expert system holds the more efficient it is.

2. Development of a diagnostic expert system in Pakistan

Due to the advantages and the need of an expert system we are developing an interview based autism diagnostic expert system 'PCADEX' abbreviated as Pakistan Childhood Autism Diagnostic Expert System. The knowledge base of the expert system is based on the expert's knowledge of the particular domain.

We used ATEC questionnaire with the different approach. We have incorporated the artificial intelligence techniques on it and molded the questions in a way that the number of questions being asked depends on the category (normal, borderline, mild to severe, severe) to which the subject seems to belong i.e. the subject who is more towards autism will have to answer more questions so as to judge the level of autism correctly whereas the subject with normal characteristics will only have to answer a few set of questions and thus system concludes intelligently like a human expert.

The implementation of ATEC questionnaire was first done in Dev C++. This program was made using the hard computing i.e. values are assigned to individual options which add up to the current score according to the option selected. The program categorizes the subject as normal, borderline, mild and severe autistic. This was an interview based diagnostic system which asks the complete set of 77 questions from every user irrespective of the category (i.e. normal, borderline, mild and severe autistic) he/she belongs. This works on the scoring methodology i.e. if the score is <56.59 than the subject is normal, if score is 56.6 – 57.4 than the subject will be borderline autistic, if the score is 57.5-59 than the subject is mild autistic and if the score is > 59 than the subject is highly autistic.

3. Development of PCADEX using PROLOG

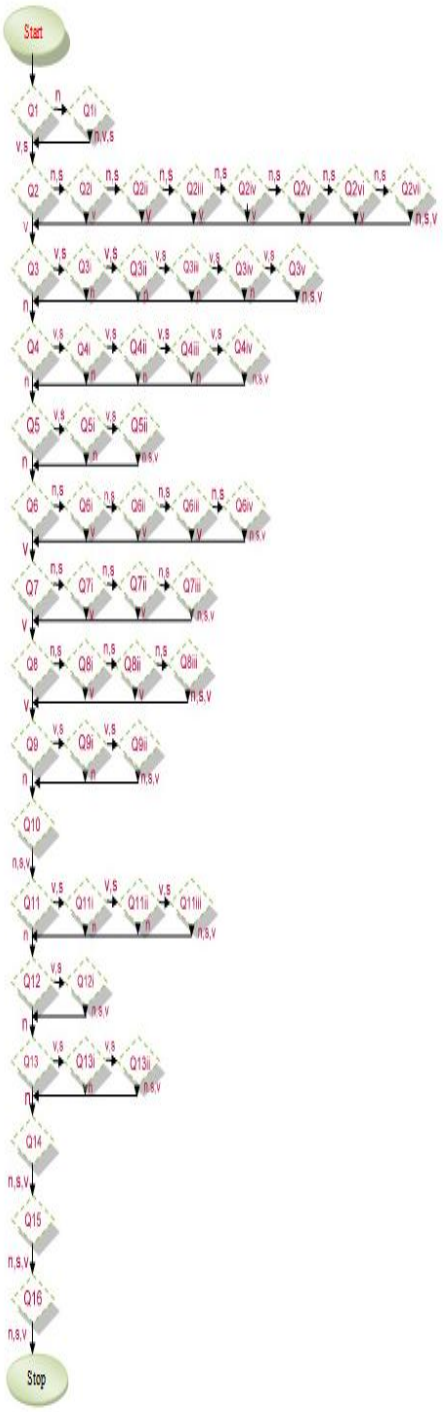
Prolog is the highest level general purpose language involving human reasoning. It is declarative language because all program statements are definitional. It strongly emphasizes on the logical relations between the entities of any given problem. The relationships between the entities/data members are defined then and on there basis the system decides the way to solve a problem. The logical program consists of explicit facts and rules to define relations, on the basis of which the implicit knowledge can be gained. This language is majorly used for data base interfaces, expert systems, and mathematical theorem provers. This language best suited our problem as we were developing an expert system which required declarative programming in which we can define our problem in the form of rules.

The ATEC questionnaire was the baseline of our project. We organized the set of questions in a way, forming blocks for the same type of questions with one

major and varying minor question and then designed the rules on the basis of blocks. From the 77 questions of the ATEC criterion we merged the questions with same meaning and reduced them to a set of 54 questions with 16 major and 38 minor questions. Each question is comprised of three options that the user can opt 'N', 'S', 'V' for not true, somewhat true and very true respectively. Rules were made applying permutations and combinations for all the questions and their respective options, using the key provided by the expert (clinical psychological). Four different paths were created that draws a conclusion that whether the subject has autism or not and if yes then to which category it may belongs to i.e. borderline, mild or severe autism.

The system works with its integrated knowledge base in PROLOG. The knowledge base contains a set of rules and facts on the basis of which the inference engine draws a conclusion. The knowledge base of the expert system contains the facts that are based on the expert's knowledge. From those facts the rules were created in which we defined four different paths using statistical techniques i.e. permutations and combinations. In which all the possible options for a question are linked to all the options of next question creating four pathways from the dataset.

The pathways generated using rules provides the flexibility to the user, as it allows the reduction and expansion of the set of questions being asked, depending on the user's data input. The system will ask a minimum set of 16 questions for a normal subject, it decides on the basis of initial input to which path it has to lead. Whereas the number of questions increases as the severity level increases as the number of questions being asked are directly proportional to the subject's autism severity level.



The flowchart in fig 2 is the representation of our system. The problem that was required to be solved is implemented successfully, using the automata, rules, and flowcharts into the PROLOG program that diagnoses the subject of being autistic or not and places him the respective category according specification provided.

4. Conclusion

The objective behind this project was to develop an automated system for autism diagnosis that can overcome the issues faced in Pakistan. This has been achieved successfully with the development of PCADEX. After the system was developed it was further tested and validated with the real subjects to determine its accuracy of results.

5. References

[1] Roberto Tuchman, MD. Autism, *Neurologic Clinics* , 2003

[2] Lisa Jo Rudy. *Causes of autism: Recent Findings, About.com: Autism*, 2006

[3] Joseph Piven. *The biological basis of autism, Current opinion in neurobiology*, 1997

[4] Sarah J Spence. *The Genetics of Autism, seminars in pediatric neurology*, 2004

[5] Bailey A , et al. *Autism as a strongly genetic disorder: Evidence from a British twin study*, Psychol Med, 1995

[6] Steffenburg S, et al, *A twin study of autism in Denmark, Finland, Iceland, Norway and Sweden*, J Child Psychol Psychiatry, 1989

[7] Christopher W. Bartlett , et al, Three autism candidate genes, International Journal of developmental neuroscience, 2005

[8] Abha R Gupta & Matthew W. State, *Recent advances in genetics*, Biol psychiatry, 2007

[9] Paul El-Fishawy, et.al. *The genetics of autism: key issues, recent findings & clinical implications*, Psychiatr Clin, 2010

[10] Gra´inne M. McAlonan, et al. *Mapping the brain in autism. A voxel-based MRI study of volumetric differences and intercorrelations in autism, Brain*, 2004

[11] Jay Schulkin. *Autism and the amygdala: An endocrine hypothesis, Science Direct: Brain and Cognition*, 2007

[12] Nancy J. Minshew & Diane L. Williams. *The New Neurobiology of Autism: Cortex, Connectivity, and Neuronal Organization*, NIH: Arch Neurol., 2007

[13] Edwin H. Cook,et.al. *Autism or Atypical Autism in Maternally but Not Paternally Derived Proximal* 15q Duplication, Am. J. Hum. Genet, 1997

[14] Elissar Andaria , et al. *Promoting social behavior with oxytocin in highfunctioning autism spectrum disorders*, PNAS, 2010

[15] Eric Hollander, et al. *Oxytocin Increases Retention of Social Cognition in Autism,* Biol Psychiatry , 2007

[16] Dr. Francisco Monegro . *NEURO-BEHAVIORAL MODEL: Autism and Brain Plasticity, QSAC clinical blog*, 2007

[17] John R. Hughes . *A review of recent reports on autism: 1000 studies published in 2007, Epilepsy & Behavio*r, 2008

[18] James L. McClelland. *The Basis of Hyperspecificity in Autism: A preliminary Suggestion Based on Properties of Neural Nets*,2000

[19] Lonnie Zwaigenbaum, et al. *Pregnancy and Birth Complications in Autism and Liability to the Broader Autism Phenotype*, J. Am. Acad. Child Adolesc. Psychiatry, 2002

[20] Monali Chowdhury, et al. *Changes in Restricted Repetitive Behaviors with age:A study of high-functioning adults with Autism Spectrum Disorders, Research in Autism Spectrum Disorders* , 2010

[21] Lidia Gabis et.al. Autism and epilepsy: *Cause, consequence, comorbidity, or coincidence?, Science Direct:Epilepsy & Behavior*, 2005

[22] C. Gillberg, et al.*Early detection of autism: Diagnostic instruments for clinicians, European Child & Adolescent Psychiatry*, 1996

[23] Lennart Gustafsson , et al. *Self-Organization of an Artificial Neural Network Subjected to Attention Shift Impairments and Familiarity Preference, Characteristics Studied in Autism*,

Journal of Autism and Developmental Disorders, 2004.

[24] Ira L. Cohen . *A Neural Network Approach to the Classification of Autism* , Journal of Autism and Developmental Disorders, 1993

[25] Tony Florio , et al. *Providing an Independent Second Opinion for the Diagnosis of Autism Using Artificial Intelligence over the Internet*, cph journal, 2009

[26] Henrik Hautop Lund . *Modular playware as a playful diagnosis tool for autistic children*, Rehabilitation Robotics, 2009 IEEE International Conference, 2009

[27] K. Arthi , et al. *Prediction of autistic disorder using neuro fuzzy system by applying ANN technique*, Int. J. Devl Neuroscience, 2008

[28] Karthik Srinivasan et al. *Exploration Of Autism Using Expert Systems*, IEEE International Conference on Information Technology, 2007

[29] Kiah Bertoglio. *New Developments in Autism*, Psychiatr Clin N Am, 2009

[30] John R. Hughes. *Update on autism: A review of 1300 reports published in 2008*, Science Direct:Epilepsy & Behavior, 2009

[31] Anthony Bailey, et al. *Autism: Towards an Integration of Clinical, Genetic, Neuropsychological, and Neurobiological Perspectives*, Journal of Child Psychology and Psychiatry, 2006

[32] Leo Kanner. *Follow up study of eleven autistic children originally reported in 1943*, Journal of autism & childhood schizophrenia, 1971

[33] Steven Novella. *Social Factors in Autism Diagnosis, Neuroscience/Mental Health*, 2010

[34] Book:-Gerald C. Davison, et al. *Abnormal Psychology* , 9th edition, Wiley International Edition, 2004

[35] Book:- Eric J Mash & david A Wolfe, *Abnormal child psychology*, 3rd edition, International student edition, 2005

[36] Book:- Ivan Bratko, *PROLOG, programming for artificial intelligence*, third edition

[37] Book: - Joseph C. Giarratano, Gary D. Riley, *Expert systems: Principles and programming, third edition.*

Optimal Placement and Tuning of Robust Multimachine PSS via HBMO

A. Ghasemi
Technical Engineering Department,
University of Mohaghegh Ardabili,
Ardabil, Iran
E-mail: ghasemi.agm@gmail.com

H. A. Shayanfar*
Center of Excellence for Power System Automation and Operation
Iran University of Science and Technology, Tehran,Iran
E-mail: hashayanfar@yahoo.com

Mohammad S. Naderi
School of Electrical Engineering & Telecommunications, UNSW,
Sydney, Australia
E-mail: m.salaynaderi@unsw.edu.au

O. Abedinia
Islamic Azad University,
Young Researcher Club, Ardabil Branch, Ardabil, Iran
E-mail: oveis.abedinia@gmail.com

Abstract— **This paper introduces an optimal placement and tuning of robust multi-machine Power System Stabilizer (PSS) using Honey Bee Mating Optimization (HBMO). This problem is formulated as an optimization problem, which is solved using HBMO technique. Hence the proposed approach employs HBMO to search for optimal location and parameters settings of a widely used Conventional fixed-structure lead-lag PSS (CPSS). One of the main advantages of the proposed approach is its robustness to the initial parameter settings. The effectiveness of the proposed method is demonstrated on the two-area four-machine (TAFM) system of Kundur in comparison with the Strength Pareto Evolutionary Algorithm (SPEA), Genetic Algorithm (GA) and Quantitative Feedback Theory (QFT) based tuned PSS under different loading condition. The proposed method of tuning the PSS is an attractive alternative to conventional fixed gain stabilizer design as it retains the simplicity of the conventional PSS and at the same time guarantees a robust acceptable performance over a wide range of operating and system condition.**

Keywords: Optimal Location, PSS, HBMO, CPSS, TAFM System.

I. INTRODUCTION

Power systems are inherently nonlinear and undergo a wide range of transient conditions, that results in under damped low frequency speed as well as power oscillations that are difficult to control. The generator excitation system maintains generator voltage and controls the reactive power flow using an Automatic Voltage Regulator (AVR).The role of an AVR is to hold the terminal voltage magnitude of a synchronous generator at a specified level. AVR helps to develop the steady-state stability of power systems; however transient stability became a concern for the power system operators [1]. The damping of power system oscillations plays an important role in enhancing overall system stability .In recent decades, Power System Stabilizers (PSSs) with conventional industry structure have been extensively used in modern power systems as an efficient means of damping power oscillations [2].

Conventional Power System Stabilizers (CPSSs) are designed based on linear models representing the system‘s generators operating at a certain operating point. The performance of these designed CPSSs is acceptable as long as the system is operating close to the operating point for which the system model is obtained. However, CPSSs are not able to provide satisfactory performance results over wider ranges of operating conditions. To overcome these difficulties of PSS design and location of that, intelligent optimization based techniques have been introduced [3].

Genetic Algorithm (GA) is a powerful optimization technique, independent on the complexity of problems where no prior knowledge is available. Many PSS tuning methods use GA [3-6]. These works investigated the use of genetic algorithms for simultaneously stabilization of multi-machine power system over a wide range of scenarios through power system stabilizers with fixed parameters. [7] Formulates the robust PSS design as a multi-objective optimization problem and employs GA to solve it. Improving damping factor and damping ratio of the lightly damped or un-damped electromechanical modes are two objectives. It has been shown that taking just one of the objectives into account may yield to an unsatisfactory result for another one. Also, GA is very sufficient in finding global or near global optimal solution of the problem. GA method requires a very long run time

* Corresponding Author. E-Mail Address: hshayanfar@yahoo.com (H.A.Shayanfar)

that may be several minutes or even several hours depending on the size of the system under study [3, 8].

Artificial Neural Network (ANN) is an intelligent method which is used for PSS tuning [9]. This technique has its own advantages and disadvantages. The performance of power system is improved by ANN based controller but, the main problem of these controllers are the long training time and selecting the number of layers and number of neurons in each layer [10]. Also, the proposed techniques are iterative and require heavy computation burden due to system reduction procedure. To overcome the drawbacks of above methods, HBMO is proposed to implement optimization in this study.

The TAFM system, under various system configurations and loading conditions is employed to illustrate the effectiveness of the proposed method for placement in comparison with the SPEA and GA techniques. Also, the effectiveness of the proposed method is compared via QFT [11]. Result interpretation shows that the proposed method is effective and a worthy alternative to conventional fixed gain stabilizer design as it retains the simplicity of the conventional PSS and still guarantees a robust acceptable performance over a wide range of operating and system condition.

II. PROBLEM STATEMENT

A. Two-Area Four-Machine System

Kundur's Two-Area Four-Machine (TAFM) system consisting of two fully symmetrical areas linked together by two 220 km, 230 kV transmission lines [12] is considered as case study in this paper. This power system typically is used to study the low frequency electromechanical oscillations of a large interconnected system. "Fig. 1", shows the system and its data which is available for everybody in Matlab software's demo.

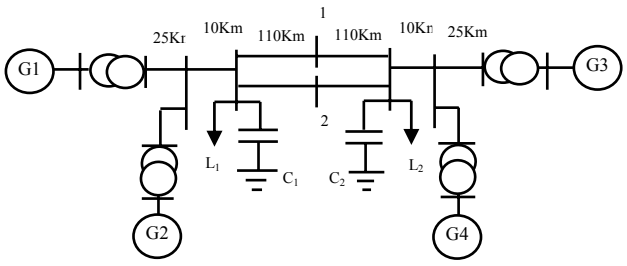


Figure 1. TAFM system.

B. PSS Design and Location

The problem of setting the parameters and locations of PSSs that assures maximum damping performance is solved using a GAs optimization procedure. For every operating condition, a linearized model is obtained. A widely used conventional lead-lag PSS is considered in this study. It can be shown as "Fig. 2".

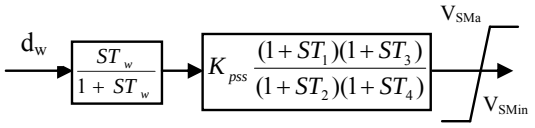


Figure 2. Power system stabilizer.

The CPSS consists of two phase-lead compensation blocks, a signal washout block, and a gain block. The PSS parameters construct the decision vector. To optimize, these parameters are experimentally limited. These limitations reduce the computation time significantly. Table I shows the up and down boundaries of the parameters.

C. HBMO Technique

The honey bee is a social insect that can survive only as a member of a community, or colony. A colony of honey bees consist of a queen, several hundred drones, 30,000 to 80,000 workers and broods during the active season. The queen is the most important member of the hive because she is the one that keeps the hive going by producing new queen and worker bees [13]. Drones' role is to mate with the queen. The HBMO Algorithm is the combination of several different methods corresponded to a different phase of the mating process of the queen. In the marriage process, the queen(s) mate during their mating flights far from the nest. In each mating, sperm reaches the spermatheca and accumulates there to form the genetic pool of the colony. The queen's size of spermatheca number equals to the maximum number of mating of the queen in a single mating flight is determined. When the queen mates successfully, the genotype of the drone is stored. At the start of the flight, the queen is initialized with some energy content and returns to her nest when her energy is within some threshold from zero or when her spermatheca is full. A drone mates with a queen probabilistically using an annealing function as [14]:

$$P_{rob}(Q,D) = e^{-(\Delta f)/(S(t))} \quad (1)$$

Where Prob (Q, D) is the probability of adding the sperm of drone D to the spermatheca of queen Q (that is, the probability of a successful mating); Δ(f) is the absolute difference between the fitness of D (i.e., f (D)) and the fitness of Q (i.e., f (Q)); and S(t) is the speed of the queen at time t. After each transition in space, the queen's speed, S(t), and energy, E(t), decay using the following equations:

$$S(t+1) = \alpha \times S(t)(2), \quad \alpha \in [0,1] \quad (2)$$

$$E(t+1) = E(t) - \gamma$$

Where, γ is amount of energy reduction after each transition. The flowchart of HBMO algorithm is presented in "Fig. 3", [8].

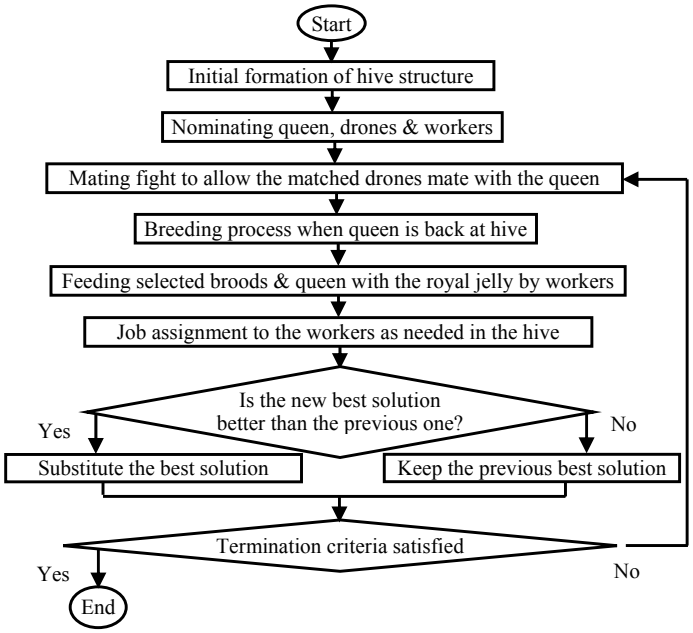


Figure 3. Algorithm and computational flowchart of HBMO.

Thus, HBMO algorithm may be constructed with the following five main stages [15]:

- The algorithm starts with the mating–flight, where a queen (best solution) selects drones probabilistically to form the spermatheca (list of drones). A drone is then selected from the list at random for the creation of broods.
- Creation of new broods by crossoverring the drones' genotypes with the queen's.
- Use of workers (heuristics) to conduct local search on broods (trial solutions).
- Adaptation of workers' fitness based on the amount of improvement achieved on broods.
- Replacement of weaker queens by fitter broods.

III. Applying HBMO Algorithm to TAFM System

To increase the system damping to the electromechanical modes and find appropriate location of PSSs the objective function *f* defined below is proposed:

$$f = \sum_{n=1}^{N_{load}} \left\{ \int_0^{t_{sim}} \left\{ |\omega_{G1} + \omega_{G2} + \omega_{G3} + \omega_{G4}| \right\} dt \right\}$$

Where, N_{load} is the load conditions and ω is the speed deviation of the generators.

TABLE I. CPSS Limitations

Parameters	T_1	T_2	T_3	T_4	V_{SMax}	K_{pss}
Lower limit	0.01	0.01	0.01	0.01	0.05	10
Uper limit	1	1	10	10	0.5	100

A. Application of HBMO to Optimum PSS Location

In this step, the proposed technique is applied to TAFM system. Two PSSs with different settings are installed at G_2 and G_4 while G_1 and G_3 are left without PSS. However, in [8] the best location of PSSs is indicated in G_1 and G_4 with similar settings. Anyway, G_2 and G_4 are considered as best locations for installation of the PSSs in this study, providing a suitable discrimination between very good and moderately good PSS settings [16]. "Fig. 4", shows the trend of objective function's variation. Also, the results of optimum parameters of PSSs and the parameters which are used to compare are presented in Table II.

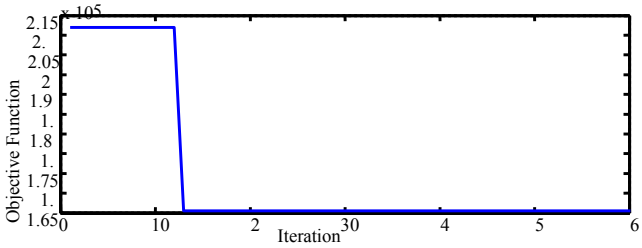


Figure 4. Objective function variation of PSSs location.

B. Application of HBMO to Robust PSS Design

The HBMO algorithm has been applied to the above optimization problem to search for optimal settings of the proposed stabilizers. The convergence rate of the objective function *f* with the number of iterations is shown in "Fig. 5". It is worth mentioning that the optimization process has been carried out with the system operating at nominal loading condition. For this step it is assumed that V_{SMax}= 0.15.

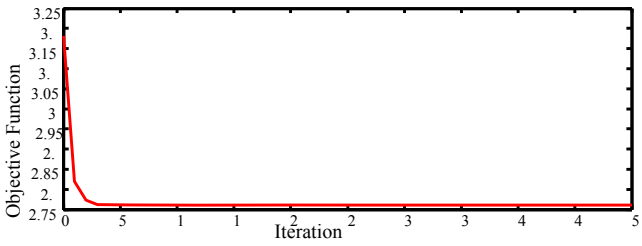


Figure 5. Objective function variation of PSSs design.

The proposed technique is applied here for design of stabilizers for 4-generator system compared with [11]. The final settings of the optimized parameters for the proposed stabilizers are given in Table III which has appropriate reaction over a wide range of loading conditions.

TABLE II. OPTIMAL PSSS' PARAMETERS OF TAFM SYSTEM

Methods		K_{pss}	T_1	T_2	T_3	T_4	V_{SMax}
HBMO	G2	41	0.001	0.002	8.4	11.8	0.36
	G4	40	0.002	0.002	9.1	12.6	0.37
SPEA	G1	45	0.26	0.01	4.2	10	0.33
	G4	45	0.26	0.01	4.2	10	0.33
GA	G1	100	0.52	0.04	0.65	5.8	0.31
	G4	100	0.52	0.04	0.65	5.8	0.31

TABLE III. THE OPTIMAL SETTINGS OF THE CONTROLLER PARAMETERS OF THE PROPOSED SCHEME

Parameters	K_{pss}	T_1	T_2	T_3	T_4
G1	20.0660	0.4020	0.6975	9.9241	4.9591
G2	21.1056	0.3426	0.6887	9.0393	4.6446
G3	22.5	0.3825	0.5191	9.6731	4.6753
G4	21.6	0.3721	0.6991	9.54	4.0432

IV. SIMULATION RESULTS

The performance of the proposed robust HBMO power system stabilizer (HBMOPSS) is evaluated by applying a large disturbance to TAFM system consist of two scenarios as:

A. *Scenario 1*

In this scenario two optimum located PSSs are installed to G_2 and G_4. To investigate the performance of the PSSs under fault conditions, 9-cycle three phase fault ground fault at bus 1 cleared without equipment have been applied to the robustness of the controllers [7, 16]. "Fig. 6", shows the variations of ω for generators in nominal operating condition. Also, "Fig. 7", shows this variation of system at light operating condition. All of these figures present large signal stability of the test system with optimum PSSs. Also, it is clear that HBMO PSSs with optimum parameters and new location of PSSs, have a better performance rather than other controllers compared with [8] in different load conditions.

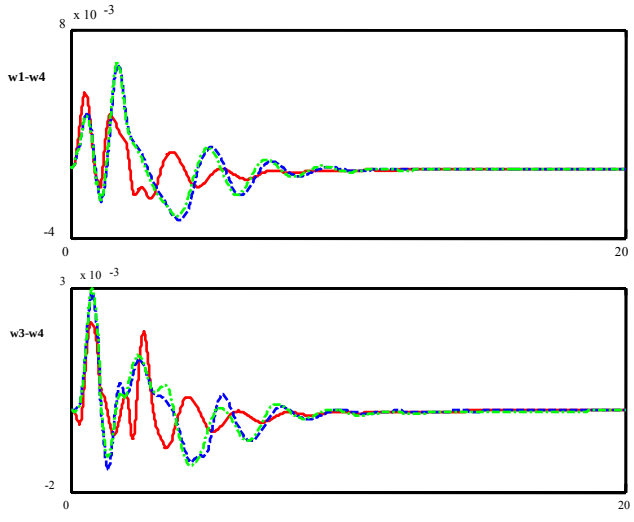


Figure 6. System response under scenario 1 with nominal loading condition: Solid (HBMOPSS) Dashed (SPEAPSS) Doted (GAPSS).

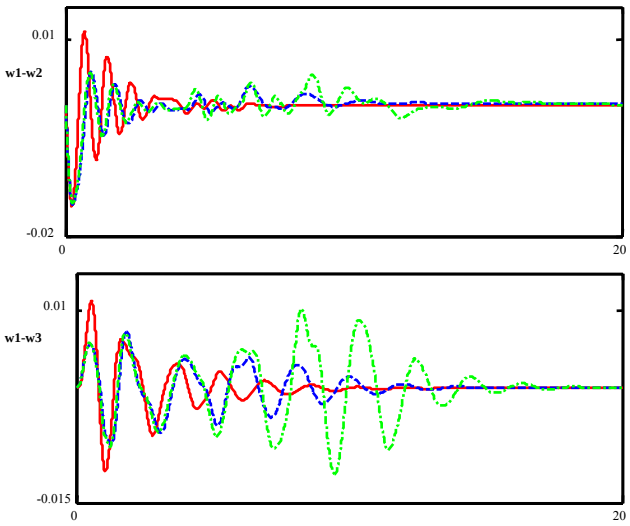


Figure 7. System response under scenario 1 with light loading condition: Solid (HBMOPSS) Dashed (SPEAPSS) Doted (GAPSS).

B. *Scenario 2*

This scenario is applied to TAFM system which is compared with QFT technique [11]. In this step, four PSSs are installed in the proposed system. Time responses of the resulting closed-loop system with all four generators fitted with stabilizers were simulated for various disturbances and operating conditions. "Fig. 8", shows the response of the system with all generators fitted with stabilizers and at heavy operating condition to a small impulse disturbance in the voltage reference of generator 1. Furthermore, "Fig. 9", shows the response of the system at nominal operating condition to the proposed disturbance.

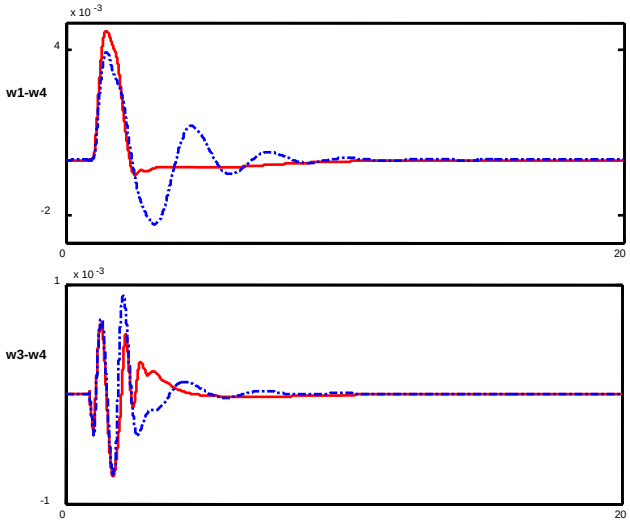


Figure 8. System response under scenario 2 with heavy loading condition: Solid (HBMOPSS) Dashed (QFTPSS).

#

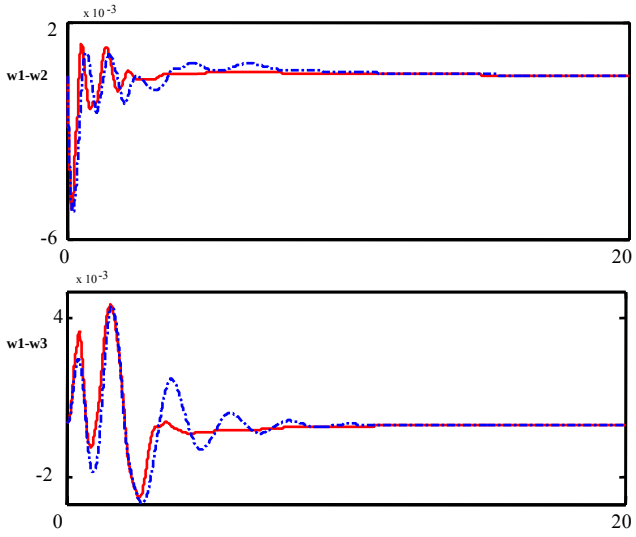


Figure 9. System response under scenario 2 with nominal loading condition: Solid (HBMOPSS) Dashed (QFTPSS).

V. CONCLUSION

In this study, a method of CPSS design named Honey Bee Mating Optimization (HBMO) was introduced for power system stabilizers' design and location. HBMO algorithm proposed in this paper is fairly easy to implement without additional computational complexity. Also, the convergence speed of this algorithm is high significantly. The described algorithm allows simultaneously tuning of PSSs and finding out their optimal locations in TAFM system. The effectiveness of the proposed method is tested on 4-machines 10-buses Kundur's system for a wide range of load demands and disturbances under different operating conditions. The proposed method is compared with SPEA, AG and QFT techniques. The simulation results confirm that the proposed HBMOPSS can work effectively over a wide range of loading conditions and seems to be superior to the mentioned intelligent methods.

REFERENCES

[1] M. J. Basler, R. C. Schaefer, "Understanding power system stability", IEEE Trans. Industry Applications, Vol. 44, pp. 463-474, 2008.

[2] K. R. Padiyar, "Power system dynamics, stability and control", Second Ed., 2006.

[3] O. Abedinia, M. Salay Naderi, A. Jalili, B. Khamenehpour, "Optimal tuning of multi-machine power system stabilizer parameters using genetic-algorithm", Proceedings of International Conference on Power System Technology, pp. 1-6, Hangzhou, China, 24-28 October, 2010.

[4] L. Davis, "Genetic algorithms and simulated annealing", Pitman, 1987.

[5] H.A. Shayanfar, H. Shayeghi, O. Abedinia, A. Jalili, "Design rule-base of fuzzy controller in multi-machine power system stabilizer using genetic algorithm", In Proceedings of the International Conference on Artificial Intelligence, Las Vegas, Nevada, USA, pp. 43-49, July 2010.

[6] P. Zhang, A.H. Coonick, "Coordinated synthesis of PSS parameters in multimachine power systems using the method of inequalities applied to genetic algorithms", IEEE Transactions on Power Systems, Vol. 15, pp. 811-816, 2000.

[7] I. Kamwa, R. Grondin, G. Trudel, "IEEE PSS2B versus PSS4B: the limits of performance of modern power system stabilizers", IEEE Transactions on Power Systems, Vol. 20, 2005.

[8] H. Yassami, A. Darabi, S.M.R. Rafiei, "Power system stabilizer design using Strength Pareto multi-objective optimization approach", Electric Power Systems Research, Vol. 80, pp. 838-846, 2010.

[9] A.L. Barreiros, M.D. Ferreira, T. Costa-Jr, W. Barre-Jr, A.P. Lopes, "A neural power system stabilizer trained using local linear controllers in a gain-scheduling scheme", Elect. Power Energy Syst., Vol. 27, pp. 473-479, 2005.

[10] M. Caner, N. Umurkan, S. Tokat, S.V. Ustun, "Determination of optimal hierarchical fuzzy controller parameters according to loading condition with ANN", Expert Systems with Applications, Vol. 34, pp. 2650-2655, 2008.

[11] P.S. Rao, E.S. Boje, "A quantitative design approach to PSS tuning", Electric Power Systems Resarch, Vol. 73, pp. 249-256, 2005.

[12] P. Kundur, "Power System Stability and Control", McGraw-Hill, New York, 1994.

[13] H.O. Bozorg, A. Afshar, M.A. Marino, "Honey-Bees Mating Optimization (HBMO) algorithm: A new heuristic approach for water resources optimization", Water Resources Management, Vol. 20, pp. 660-680, 2006.

[14] H.A. Abbass, "Marriage in honey-bee optimization : A haplometrosis polygynous swarming approach", The Congress on Evolutionary Computation, Vol. 1, pp. 207-214, 2001.

[15] A. Arefi, MR. Haghifam, S.H. Fathi, T. Niknam, J. Olamaei, "A novel algorithm based on honey bee mating optimization for distribution harmonic state estimation including distributed generators", IEEE Bucharest Power Tech Conference, pp. 1-7, July 2009.

[16] Y.L. Abdel-Magid, M.A. Abido, "Optimal multi-objective design of robust power system stabilizers using genetic algorithms", IEEE Transactions on Power Systems, Vol. 18, pp. 1125–1132, 2003.

BIOGRAPHIES

Ali Ghasemi received the B.S. degree in Electrical Engineering from Esfahan University of Technology, Esfahan, Iran in 2009. Currently, he is a M.S.E. student in Technical Eng. Department of the University of Mohaghegh Ardabili, Ardabil, Iran. His areas of interest are in the Application of Heuristic Optimization to Power System Control.

Heidarali Shayanfar received the B.S. and M.S.E. degrees in Electrical Engineering in 1973 and 1979, respectively. He received his Ph. D. degree in Electrical Engineering from Michigan State University, U.S.A., in 1981. Currently, he is a Full Professor in Electrical Engineering Department of Iran University of

Science and Technology, Tehran, Iran. His research interests are in the Application of Artificial Intelligence to Power System Control Design, Dynamic Load Modeling, Power System Observability Studies, Voltage Collapse, Congestion Management in a Restructured Power System, Reliability Improvement in Distribution Systems and Reactive Pricing in Deregulated Power Systems. He has published more than 350 technical papers in the International Journals and Conferences Proceedings. He is a member of Iranian Association of Electrical and Electronic Engineers and IEEE.

Mohammad Salay Naderi received his B.Sc. in electric power engineering (1996) and M.Sc. (1998) in high voltage engineering from Sharif University of Technology in Tehran, Iran. In 2006, he finished his PhD career at the same university which was on partial discharge monitoring and localization within power transformer. He joined Energy Systems Research Group in the School of Electrical Engineering and Telecommunications at UNSW in 2004 as Visiting Research Associate and continued his research until late 2005. He worked with Iran Power Generation and Transmission Company (Tavanir Holding Co.) as Deputy General Manager of Transmission System Technical Administration Bureau from 2005 to 2010. In 2010 he joined the School of Electrical Engineering and Telecommunications at UNSW as Lecturer. His research and teaching interests are in high voltage engineering and condition monitoring of power equipment; sustainable energy systems analysis and development as well as distributed generation and its effects of power system protection.

Oveis Abedinia received the B.S. and M.Sc. degrees in Electrical Engineering from Azad University, Ardabil and Science and Technology Research Branch, Tehran, Iran in 2005 and 2009, respectively. His areas of interest in research are the Application of Robust Control, Artificial Intelligence to Power System Control and Restructuring, Optimization in Distribution Generation and Power System.

Analyzing the Classifier using the Knowledge Hierarchy

Yan Zhiyong and Xu Congfu
Department of Computer Science, Zhejiang University, Hangzhou, 310027, China
xucongfu@zju.edu.cn

Abstract - *In this paper, we analyze the classifier in machine learning using the knowledge hierarchy in information science. An instance provides the type of information answering questions begin with where and what. A classifier is the type of knowledge answering the question begins with how, and a classification model is the type of knowledge answering the question begins with why. A classifier utilizes information contained in training data set to obtain the knowledge represented by a classification model. We analyze types of information utilized by four typical classifiers, C4.5 algorithm, Naive Bayes classifier, support vector machine and k nearest neighbors. We also analyze typical improving methods of above four classifiers from the perspective of information utilized and divide information into global information and local information according to numbers of instances and attributes used.*

Keywords: Classifier, Knowledge Hierarchy, Information

1 Introduction

Classification is an important technique in machine learning. The classification problem is as follows: learning from training data set $D_{\text{train}}=\{(\boldsymbol{x}_i, y_i)|\boldsymbol{x}_i \in \boldsymbol{X}, y_i \in Y\}$, and then making predictions for unlabeled instances in test data set $D_{\text{test}} = \{\boldsymbol{x}_i|\boldsymbol{x}_i \in \boldsymbol{X}\}$. A classifier is an algorithm to solve the classification problem. There are many classifiers, and literature [1] is a good survey of existing classifiers. Four typical classifiers which will be analyzed in this paper are C4.5 algorithm [2], Naive Bayes classifier [3], support vector machine (SVM) [4] and k nearest neighbors (KNN) algorithm [5]. A classifier is usually trained on D_{train} to obtain a classification model in training phase, and then in test phase the classification model is used to make predictions for unlabeled instances.

Researchers usually regard a classification model as a function (hypothesis) from data space $\boldsymbol{X}$ to label set Y. Then the training process can be regarded as searching hypotheses space for the best hypothesis. Some researchers regard a classification model as a group of decision boundaries in data space $\boldsymbol{X}$. Then the training process can be regarded as dividing data space $\boldsymbol{X}$ into decision regions separated by decision boundaries. Although there are above two different views, the same thing of both above two views is that a classifier utilizes information contained in training data set to construct a classification model. However, there have been no studies on what types of information classifiers utilize to construct classification models. One reason for above phenomenon is that there is no proper theory in machine learning to analyze types of information.

The knowledge hierarchy is an important theory in information science, and it was first proposed by Ackoff [6]. The knowledge hierarchy usually contains three levels, data, information and knowledge [6]. Information is contained in data, and is a higher level than data. Knowledge is contained in information, and is a higher level than information. There are many definitions of above three concepts [7], among which we adopt definitions of interrogative theory [8]. Interrogative theory makes definitions of above three concepts according to what questions they answer [8]. According to interrogative theory, data answers no special questions, information answers questions begin with *who*, *what*, *where* and *when*, and knowledge answers questions begin with *how* and *why*.

In this paper, we adopt the knowledge hierarchy to analyze the classifier. In classification, training data set is in the data level, and the classification model is in the knowledge model. The classifier can also be regarded as in knowledge level. We propose definitions of data, information and knowledge in classification according to interrogative theory. We analyze types of information utilized by four typical classifiers, C4.5 algorithm, Naive Bayes classifier, SVM and KNN, and analyze typical improving methods of above four classifiers from the perspective of information.

2 The knowledge hierarchy in classification

In this section, we will first present the knowledge hierarchy in classification, and then analyze decision tree algorithms and sequential minimal optimization (SMO) algorithm [9] using doubly linked chain [10].

2.1 The knowledge hierarchy in classification

The training process of a classifier can be regarded as *classifier + training data = classification model*. In above formula, *training data* belongs to the data level, while *classifier* and *classification model* belong to the knowledge level. A classifier is a method proposed by researchers to obtain a classification model from training data set and make predictions, and therefore, it is knowledge answering the question begins with *how*. A classification model is knowledge generated by a classifier in a special classification problem, and it can be regarded as a function or a group of

decision boundaries. A classification model is the basis according to which a classifier makes predictions. Therefore, a classification model is knowledge answering the question begins with *why*.

Then we discuss what information training data set provides for classifiers. Training data set contains many instances, which consist of several attributes and labels. Classifiers do not consider meanings of attributes, and what they consider is only the relative location of an instance in data space $\boldsymbol{X}$. Another factor a classifier considers is labels of instances. Therefore, information that an instance provides for a classifier consists of its location in data space $\boldsymbol{X}$ and its label. Thus, an instance contains information answering questions begin with *where* and *what* for a classifier.

Finally, we present definitions of data, information and knowledge in classification.

Definition 1 (*data*): In classification, data consist of attributes and labels.

Definition 2 (*information*): In classification, information provided by an instance $(\boldsymbol{x}, y)$ is that on location $\boldsymbol{x}$ in data space $\boldsymbol{X}$, there is an instance with label y, and it answers questions begin with *where* and *what*.

Definition 3 (*knowledge*): In classification, a classifier is a method of obtaining a classification model from training data set and then using the classification model to make predictions, and it is knowledge answers the question begins with *how*. The classification model is the basis according to which a classifier makes predictions, and it is knowledge answers the question begins with *why*.

The knowledge hierarchy in classification is shown in Figure 1.

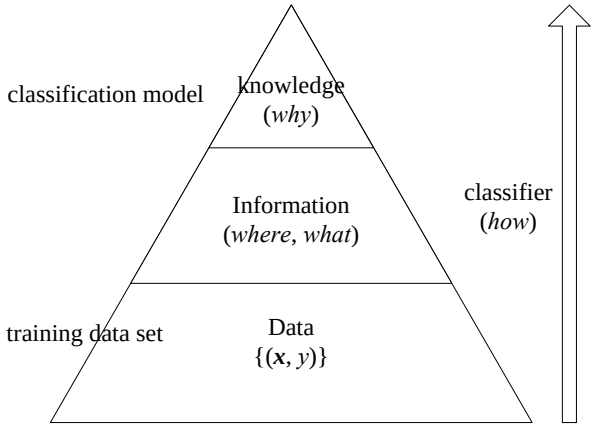


Figure 1 The knowledge hierarchy in classification

According to above definitions, we have the following formula K[*how*] + I[(*where*, *what*)] = K[*why*]. Information in definition 2 is provided for all classifiers, and besides it, a classifier can obtain more information from training data set. Tuomi proposed the reversed knowledge hierarchy [11]. According to the reversed knowledge hierarchy, only when there has been knowledge there will be information. A classifier is a kind of knowledge. According to Tuomi's theory, there is information special to a classifier. For example, Naive Bayes classifier utilizes conditional probability information. If a classifier utilizes more information, it may obtain better classification ability.

According to formula K[*how*] + I[(*where*, *what*)] = K[*why*], if a classifier provides more knowledge, it many also obtain better classification ability. Researchers study incorporating prior knowledge into SVM [12], which can improve the classification ability of SVM.

A classification model is a kind of knowledge, and the comprehensibility is an important characteristic for it. For example, the comprehensibility of SVM is very poor, so researchers perform studies on extracting rules from trained SVM to improve the comprehensibility [13].

2.2 Doubly linked chain

Literature [10] proposes the double linked chain of knowledge hierarchy, which extands studies of literature [11]. We adopt the doubly linked chain of literature [10] to propose the doubly linked chain in classification, which is shown in Figure 2.

$$\begin{array}{ccc} \text{K}[how] + \text{I}[(where, what)] & = & \text{K}[why] \\ \| & & + \\ D_{\text{train}} & & \end{array}$$

Figure 2 The doubly linked chain in classification

From Figure 2, we obtain the process of training classifiers iteratively, which is shown in Figure 3.

$$\begin{array}{ccc} \text{K}[how] + \text{I}[(where, what)] & \Rightarrow & \text{K}[why] \\ & \wedge & \\ \| & & + \\ D_{\text{train}} & & \end{array}$$

Figure 3 The process of training classifiers iteratively

From Figure 3, a classifier first utilizes information contained in training data set D_{train} to generate a classification model CM_i, then new information being obtained by CM_i and D_{train}, and finally the classifier utilizes new information to obtain new classification model CM_{i+1}. By iteratively generating and utilizing new information, a classifier can generate a more appropriate classification model. Decision tree algorithms and SMO algorithm are two examples.

Decision tree algorithms first utilize information contained in the whole training data set to select an attribute as the root node, and partition data set into several data sub-set. The root node can be regarded as a simple classification model, which in named CM_0. Then decision tree algorithms use CM_0 to generate new information, which is contained in data sub-sets generated by CM_0. Decision tree algorithms partition data sub-sets to generate children nodes of the root node, which can be regarded as that new information is utilized to obtain a new classification model CM_i.

SMO algorithm is proposed to obtain alpha values $\{\alpha_i\}$. SMO algorithm first initializes, and then iteratively tunes alpha values. SMO algorithm first obtains a classification model CM_0, which corresponds to initial alpha values. Then new information is generated by CM_0. SMO algorithm utilizes

new information to obtain a new classification model CM_i. Above process is repeated iteratively to obtain an more appropriate classification model.

Both decision tree algorithms and SMO algorithm confront the same problem, the stopping criteria. The stopping criteria determines when a classifier stops generating and utilizing new information.

The thinking can be applied to all classifiers. The SD algorithm can be regarded as an attempt [14]. The SD algorithm is proposed to train classifiers in decision regions of Naive Bayes classifiers. Naive Bayes classifier is trained in the first iteration, and its classification model is CM_0. CM_0 generates new information, which is contained in decision regions of CM_0. Classifiers trained in decision regions of Naive Bayes classifier utilize above new information to obtain the new classification model CM_1.

3 Analyzing classifiers using the knowledge hierarchy

In this section, we first discuss classification models of four typical classifiers, and then discuss types of information they utilize. Four classifiers are C4.5 algorithm, Naive Bayes classifier, SVM and KNN.

Classification models of former three classifiers can be found in literature [15]. Therefore, in this section, we only discuss the classification model of KNN. In training phase, although KNN does not construct any classification model, the classification model of KNN exists. The classification model of KNN consists of several super polyhedrons. The decision regions of KNN are super polyhedrons, and the decision boundaries are super polyhedral surfaces.

C4.5 algorithm generates a decision tree from top to down in training phase, and on generating each branch of tree it utilizes information gain ratio of instances. Information gain is a description of instances, and it does not consider accurate location of instances. When generating a new branch, C4.5 algorithm divides data set into several small data sets. Each leaf node corresponds to a small data set. There are usually more than one instance in each small data set. If a small data set has three instances $(\boldsymbol{x}_a, y_a)$, $(\boldsymbol{x}_b, y_b)$ and $(\boldsymbol{x}_c, y_c)$, and the parent node branches on attribute $\boldsymbol{x}[i]$, under assumptions that $\boldsymbol{x}_a[i] < \boldsymbol{x}_b[i] < \boldsymbol{x}_c[i]$, the accurate location of $\boldsymbol{x}_b[i]$ has no impact on the training result as far as it lies in interval $(\boldsymbol{x}_a[i], \boldsymbol{x}_c[i])$. C4.5 algorithm checks attributes one by one, and does not consider relative values among attributes. Therefore, it does not make effect on C4.5 algorithm to scale some attributes of data set.

Naive Bayes classifier first estimates conditional probabilities of attributes on labels, and then makes predictions according to conditional probabilities. Naive Bayes classifier transforms information contained in training data set answering questions begin with *where* and *what* into conditional probabilities. Conditional probabilities are properties of all instances. Therefore, changes of few instances may not affect the classification result. Naive Bayes classifier assumes attributes are independent on class. Thus, it does not make effect on Naive Bayes classifier to scale some attributes of data set.

SVM calculates the decision boundary by a few instances near the decision boundary, which are named support vectors. For hard margin SVM, training SVM on only support vectors can obtain the same classification model as training SVM on all instances, which means non support vectors provide no information for hard margin SVM. For soft margin SVM, SVM considers the training accuracy of obtained classification model, which means SVM utilizes information contained in non support vectors. SVM utilizes information of accurate locations and labels of support vectors. Besides support vectors, soft SVM utilizes information of non-accurate location and label of non support vectors, which means that changes of locations of non support vectors can not affect the training result as far as non support vectors do not become support vectors. SVM calculates inner product of two instances in high dimension feature space. Thus, unlike C4.5 algorithm and Naive Bayes classifier, it makes effect on SVM to scale some attributes of data set. Therefore, it is natural for SVM to normalize data. However, normalization may result in information loss.

We compare hard and soft margin SVMs trained on whole data set and only on support vectors. The kernel function used is quadratic polynomial kernel. The data set is *breast-cancer* of UCI data set [16]. Test accuracies are obtained by 10 times 10-fold cross validation. Test accuracies of hard margin SVMs trained on whole data set and support vectors are both 70.75%, while two accuracies of soft margin SVMs are 69.14% and 67.10% respectively. Thus, soft margin SVM utilizes more information than hard margin SVM.

KNN utilizes information of locations and labels of only instances near decision boundaries. Information contained in instances far from decision boundaries has no impact on classification results. One shortage of KNN is that it has to store all training instances, although some of them have no effect on the classification result. There have been many studies on reducing training instances storage while not reducing classification ability of KNN [17]. Literature [18] proposed Iterative Case Filtering algorithm to reduce 80% training instances storage. KNN need calculate the distance between two instances, so it makes effect on KNN to scale some attributes of data set.

It is a common sense that there is no classifier can outperform other classifiers in all classification problems. One reason is that types of information utilized by different classifiers are not the same, and training data set in a classification problem may contain more information of certain type, so a classifier utilizing this type of information will obtain very good classification ability in the classification problem.

From above discussion, scaling affects SVM and KNN, but does not affect other two classifiers. We then perform an experiment on *iris* of UCI data sets [16] to illustrate this.

Data set *iris* contains four attributes, sepal length, sepal width, petal length and petal width, all of which are measured

by centimeters (cm). Descriptions and some instances of data set *iris* are listed in Table 1.

We measure former two attributes by millimeters (mm), and then values of these two attributes are scaled by 10. Descriptions and some instances of transformed data set *iris* of Table 1 are listed in Table 2.

We perform 10 times 10-fold cross validation for C4.5 algorithm, Naive Bayes classifier (NB), SVM and KNN on original and transformed data sets of *iris*. Test accuracies of above four classifiers are listed in Table 3.

From Table 3, test accuracis of C4.5 and Naive Bayes classifier of two data sets are the same, while test accuracies of SVM and KNN of two data sets are different. Experimental results indicate that scaling affects classification abilities of only SVM and KNN, which validates above discussion.

Table 1 Descriptions of original data set *iris* and some instances

instance no.	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	label
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa

Table 2 Descriptions of transformed data set *iris* and some instances

instance no.	sepal length (mm)	sepal width (mm)	petal length (cm)	petal width (cm)	label
1	51	35	1.4	0.2	Iris-setosa
2	49	30	1.4	0.2	Iris-setosa
3	47	32	1.3	0.2	Iris-setosa

Table 3 Test accuracies (%) of four classifiers on original and transformed data set *iris*

Data Set	C4.5	NB	SVM	KNN
original	94.33±0.96	95.73±0.47	97.80±1.09	96.07±0.21
transformed	94.33±0.96	95.73±0.47	98.00±0.31	85.20±1.50

4 Information utilized by methods of improving classifiers

In section 3, we have analyzed information utilized by four typical classifiers. In this section we will first discuss some methods of improving above four typical classifiers by utilizing more information. Then we will divide information into global information and local information, and discuss how improving methods utilize local information to improve generalization abilities of original classifiers.

4.1 Information utilized by typical methods of improving four classifiers

For C4.5 algorithm, famous improving methods are ensemble learning methods, for example bagging [19]. Bagging trains classifier C many times to obtain many classification models. In each iteration, bagging samples from training data set to obtain a new training data set of the same size. New training data sets change distribution of training data. According to the discussion in section 3, C4.5 algorithm does not utilize information of the accurate location of $(\boldsymbol{x}_b, y_b)$. By generating new training data sets, bagging may make C4.5 algorithm utilize information of accurate locations of instances like $(\boldsymbol{x}_b, y_b)$. In this sense, Bagging makes C4.5 algorithm utilize more information.

For Naive Bayes classifier, one typical improving method is NBTree [20]. NBTree generates a decision tree, and trains a Naive Bayes classifier on each leaf node of the decision tree. Literature [20] regards that NBTree relaxed the conditional independent assumption of Naive Bayes classifier. We regard that NBTree makes Naive Bayes classifier utilizes more information contained in instances near decision boundaries.

For SVM, Maxi-min margin machine (M^4) [21] is a typical improving method. Literature [21] found that SVM does not utilize information contained in instances far from decision boundaries, which are usually non support vectors. M^4 utilizes co-variance matrix, which is a type of global information calculated by all instances.

For KNN, KNC [22] is a typical improving method. Like SVM, KNN does not utilize information contained in instances far from decision boundaries. C4.5 algorithm, Naive Bayes classifier and soft margin SVM utilize above information in different degree. KNC is proposed to combine KNN with other three classifiers. Experimental results on 20 UCI data sets demonstrate that KNC improve classification ability of KNN and other three classifiers.

4.2 Global information and local information

Literature [23] divides algorithms of machine learning into global learning and local learning. Global learning

algorithms make descriptions of training data sets, while local learning algorithms obtain hyper-surfaces to separate instances with different labels. Naive Bayes classifiers belongs to global learning, while SVM belongs to local learning.

In this paper, we divide information into global information and local information according to numbers of instances and numbers of attributes from which information is obtained. Definitions are as follows.

Definition 4 (*global information*): Global information is information obtained from all instances or all attributes.

Definition 5 (*local information*): Local information is information obtained from a part of instances or a part of attributes.

Global information and local information are relative concepts. Other conditions being the same, information obtained from all instances is called global information, while information obtained from a part of instances is called local information. Other conditions being the same, information obtained from all attributes is called global information, while information obtained from a part of attributes is called local information.

Majority voting classifier (MVC) is a simple classifier, which makes the label with the most instances as the unique prediction. The unique prediction of MVC is in equation (1).

$$MVC(x) = \arg\max_{Y_i} \left| \{(x_j, y_j) \mid (x_j, y_j) \in D, y_j = Y_i\} \right|. \quad (1)$$

From equation (1), from the perspective of the number of instance, MVC utilizes global information, while from the perspective of the number of attributes, MVC utilizes local information, because it utilizes no attributes.

Decision tree algorithms and KNN are two algorithms of improving MVC. Areas corresponding to leaf nodes of a decision tree being denoted as $\{A_i\}$, instances set in A_i being denoted as D_i, the prediction of a decision tree algorithm is in equation (2).

$$DT(x) = \arg\max_{Y_i} \left| \{(x_j, y_j) \mid (x_j, y_j) \in D_k, \{x\} \bigcap R_k \neq \varnothing, y_j = Y_i\} \right|. \quad (2)$$

Comparing equation (1) with equation (2), decision tree algorithms can be regarded as improving methods of MVC. Decision tree algorithms use an MVC to determine the label of each leaf node. Decision tree algorithms decrease the number of instances, when compared with MVC, which makes decision tree algorithms utilize local information from the perspective of the number of instances. However, from the perspective of the number of attributes, decision tree algorithms utilize global information instead of local information utilized by MVC.

The set of k nearest neighbors of instance $\boldsymbol{x}$ being denoted as $D_{KNN}(\boldsymbol{x})$, the prediction of KNN is in equation (3).

$$KNN(x) = \arg\max_{Y_i} \left| \{(x_j, y_j) \mid (x_j, y_j) \in D_{KNN}(x), y_j = Y_i\} \right|. \quad (3)$$

Comparing equation (1) with equation (2), KNN can be regarded as an improving method of MVC. KNN trains an MVC on the data sub-set composed of k nearest neighbors of instance $\boldsymbol{x}$. KNN utilizes local information from the perspective of the number of instances. However, KNN utilizes global information from the perspective of the number of attributes.

From above analysis, decision tree algorithms and KNN both utilize more information contained in training data set to obtain better generalization abilities than MVC.

From both perspectives of the number of instances and the number of attributes, decision tree algorithms utilize global information. Bagging and random subspace method (RSM) [24] improve decision tree algorithms from above two perspectives respectively. In each iteration, bagging performs sampling with replacement on training data set D to obtain new training data set D_i with size $|D|$. Different instances of D_i occupy 63.2% of D. Therefore, bagging utilizes local information instead of global information by decision tree algorithms from the perspective of the number of instances. RSM randomly selects some attributes, and then trains a decision tree algorithm on the new training data set which is the projection of training data set on selected attributes. RSM utilizes local information from the perspective of the number of attributes.

From above analysis, it will improve generalization abilities of classifiers by utilizing some types of local information.

5 Conclusions

In this paper, we adopt the knowledge hierarchy to study the classifier. We present definitions of data, information and knowledge in classification according to interrogative theory. In classification, an instance provides information answering questions begin with *where* and *what*. A classifier is a method of utilizing information of training data set to construct a classification model and using the classification model to make predictions for unlabeled instances, and it is a kind of knowledge answering the question begins with *how*. The classification model is the training result of a classifier, and it is the basis according to which a classifier makes predictions for unlabeled instances, so it is a kind of knowledge answering the question begins with *why*. The training phase of a classifier can be regarded as K[*how*] + I[(*where*, *what*)] = K[*why*].

We analyze types of information utilized by four typical classifiers, C4.5 algorithm, Naive Bayes classifier, SVM and KNN. Types of information utilized by above four classifiers are not the same. The reason why there is no classifier outperforms other classifiers is that a training data set may contain more information of a certain type utilized by a classifier. We also analyze some typical improving methods of above four classifiers from the perspective of utilizing more information. Finally we divide information into global information and local information and analyze how many classifiers utilize some types of local information to obtain better classification abilities.

6 Acknowledgement

This paper is supported by the National Natural Science Foundation of China under Grant No. 60970081, and partially by the National Basic Research Program of China under Grant No. 2010CB327903.

7 References

[1] Kotsiantis, S.B., Zaharakis, I.D., Pintelas, P.E.. "Machine Learning: A Review of Classification and Combining Techniques". Journal of Artificial Intelligence Review, 26(3): 159–190, 2006.

[2] Quinlan, J.R. "C4.5: Programs for machine learning". San Mateo: Morgan Kaufmann, 1993.

[3] Domingos, P., Pazzani, M. "On the optimality of the simple Bayesian classifier under zero-one loss". Machine Learning, 29: 103–130, 1997.

[4] Vapnik, V.N. "The nature of statistical learning theory". Berlin Heidelberg: Springer, 1995.

[5] Fix, E., Hodges, JR J.L. "Discriminatory analysis, non-parametric discrimination: consistency properties". Randolph Field, Texas: United States Air Force, School of Aviation Medicine, Technical Report 21-49-004(4), 1951.

[6] Ackoff, R.L. "From data to wisdom". Journal of Applied Systems Analysis, 16: 3–9, 1989.

[7] Zins, C. "Conceptual approaches for defining data, information, and knowledge". Journal of the American Society for Information Science and Technology, 58(4): 479–493, 2007.

[8] Quigley, E.J., Debons, A. "Interrogative Theory of information and knowledge". Proceedings of SIGCPR'99. New Orleans: ACM Press, 4–10, 1999.

[9] Platt, J.C. "Fast training of support vector machines using sequential minimal optimization". In: SCHOLKOPF B, BURGES C.J.C, SMOLA A.J., ed. Advances in kernel methods: Support vector machines. Cambridge, MA: MIT Press, 185–208, 1999.

[10] Syed, A., Shah, A. "Data, information, knowledge, wisdom: a doubly linked chain?" In: Proceedings of the 101st international conference on information and knowledge engineering, 270–8, 2006.

[11] Tuomi, I. "Data is more than knowledge: implications of the reversed knowledge hierarchy for knowledge management and organizational memory". Journal of Management Information Systems, 16(3): 103–117, 1999.

[12] Barakat, N., Bradleyb, A.P. "Rule extraction from support vector machines: A review". Neurocomputing, 74(1-3): 178–190, 2010.

[13] Lauer, F., Bloch, G. "Incorporating prior knowledge in support vector machines for classification: A review". Neurocomputing, 71: 1578–1594, 2008.

[14] Yan, Z., Xu, C., Pan, Y. "Improving Naive Bayes classifier by dividing its decision regions". Accepted by Journal of Zhejiang University-SCIENCE C (Computers & Electronics), 2011.

[15] Yan, Z., Xu, C. "Studies on Classification Models using Decision Boundaries". Proceedings of the 8th IEEE International Conference on Cognitive Informatics. Hong Kong: IEEE, 287–294, 2009.

[16] Murphy, P.M., Aha, D.W. "UCI repository of machine learning databases". Irvine, CA: University of California, Department of Information and Computer Science, 1998. http://www.ics.uci.edu/~mlearn/MLRepository.html.

[17] Olvera-Lopez, J.A., Carrasco-Ochoa, J.A., Martinez-Trinidad, J.F., et al. "A review of instance selection methods". Artificial Intelligence Review, 34: 133–143, 2010.

[18] Brighton, H., Mellish, C. "Advances in instance selection for instance-based learning algorithms". Data Mining and Knowledge Discovery, 6(2): 153–172, 2002.

[19] Breiman, L. "Bagging predictors". Machine Learning, 24(2): 123–140, 1996.

[20] Kohavi, R. "Scaling up the accuracy of Naive-Bayes classifiers: A decision-tree hybrid". Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining. Menlo Park, California: AAAI Press, 202–207, 1996.

[21] Huang, K., Yang, H., King I., et al. "Maxi-min margin machine: learning large margin classifiers locally and globally". IEEE Transactions on Neural Networks, 19: 260–272, 2008.

[22] Yan, Z., Xu, C. "Combining KNN Algorithm and Other Classifiers". In Proceedings of 9th IEEE International Conference on Cognitive Informatics (ICCI'10), Beijing, China. 800–805, 2010.

[23] Huang, K.Z., Yang, H.Q., King, I., et al. "Machine learning: modeling data locally and globally". Springer-Verlag, New York Inc, 1–28, 2008.

[24] Ho, T.K. "The random subspace method for constructing decision forests". IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(8): 832–844, 1998.

A Tool for Web Links Prototyping

Inma Hernández, Hassan A. Sleiman, David Ruiz, Rafael Corchuelo
University of Seville
Seville, Spain
Email: {inmahernandez, hassansleiman, druiz, corchu}@us.es

***Abstract*—Crawlers for Virtual Integration processes must be efficient, given that VI process is online, which means that while the system is looking for the required information, the user is waiting for a response. Therefore, downloading a minimum number of irrelevant pages is mandatory in order to improve the crawler efficiency. Most crawlers need to download a page in order the determine its relevance, which results in a high number of irrelevant pages downloaded. We propose a tool that builds a set of prototype links for a given site, where each prototype represents links leading to pages containing a certain concept. These prototypes can then be used to classify pages before downloading them, just by analysing their URL. Therefore, they are the support for crawlers to navigate through sites downloading a minimum number of irrelevant pages while reducing bandwidth, making them suitable for VI systems.**

***Index Terms*—Web Crawling, Web Page Classification, Virtual Integration, Prototype-based Classification**

I. Introduction

Virtual Integration aims at accessing web information in an automated manner, starting with a query, in which users express their interests, and obtaining information relevant to that query from the web. Automated access to the web requires a crawler, which is a tool able to navigate through web sites automatically, looking for relevant information. Traditional crawlers visit every link on every page, download their target, and check whether the page contains relevant information. This means that, even when a page is irrelevant, the crawler has to download it to realise it, which results in a large number of irrelevant pages downloaded.

Note that the VI process is online, which means that while the system is looking for the required information, the user is waiting for a response. Therefore, downloading a minimum number of irrelevant pages is mandatory in order to improve the crawler efficiency, which is a concern for several researchers [14].

We propose a classifier that help crawlers to efficiently navigate through web sites. This classifier is able to determine if a web page is relevant or not by analysing exclusively its URL.

There are some crawling techniques that improve traditional crawlers efficiency by endowing the crawler with classification skills. For example, focused crawlers [1], [10], [13], [22], [24], [25] aim at finding pages belonging to one or more topics exclusively, so they are supported by a content based classifier that determines whether each page belongs to the topics, discarding the rest. Other crawlers include classifiers based on other features, like page structure [19], [20], [27]. Our classification proposal is different, since it is based on features that are not in the page to be classified, but in pages that link to it. Therefore, it is not necessary to download a page in order to classify it, which avoids downloading irrelevant pages, reducing the bandwidth and making it efficient and suitable for VI systems.

Other crawling techniques rely completely on the user to define the navigation patterns [2], [5], [8], [23], [29]. Instead, our proposal is automated and requires a minimum intervention from the user. Furthermore, our classifier is trained using an unlabelled training set of URLs, thus relieving the user from the tedious task of assigning a label to each page in the training set.

Our experience focuses on web sites that follow a certain navigational pattern, which is the most common pattern in the Web [19]. This pattern starts with a form page that allows issuing queries; then, after users submit a query, the system returns a hub, that is, a page containing an indexed list of answers to it, each of them containing just a brief description and a link to a page with more details. Note that the term "hub" is based on the hub and authority concepts introduced by Kleinberg [18].

Hubs in this kind of web sites are often defined by populating some patterns or scripts with data stored in a database [7]. This has two main implications: first, all hubs from the same web site usually share a common template with some fixed parts. Usually, this common parts are structured in the form of headers, footers and side bars containing navigational aids, copyright information and advertising [30], which frame the page areas that contain the information that varies from hub to hub. The second implication is that hubs and detail pages are generated on demand to respond to a user query. Similarly, URLs that point to each hub and detail page in the site are generated on demand as well by the same proceeding of filling a URL pattern with keywords or numbers that identify the generated page. Therefore, all URLs from a certain site can be expressed by a collection of URL patterns.

Furthermore, our hypothesis is that there is usually a correspondence between URL patterns and the concept contained in the pages with URLs following that pattern, so that we can classify web pages containing different concepts by means of the pattern matching their URL. Therefore, our classification technique consists on finding the different URL patterns that compose links in a given web site, that is, to build a set of prototypes for all URLs in the site. Then, we use these prototypes to classify links by template matching, that is, by

assigning each link to the class associated to its matching prototype. Furthermore, our technique is able as well to detect links belonging to the web site template, so that the crawler can process them adequately.

As a motivating example, in Figure 1 we show a hub page obtained after issuing a query on Amazon.com. If the user is interested in obtaining detailed information about products, for example, following only links marked in red means avoiding to download more than 50% of the hub linked pages, with the proportional reduction of used bandwidth. We notice that all links leading to pages containing information about the concept 'Product' have a similar pattern, which is shown in the Figure as well.

The rest of the article is structured as follows. Section II introduces the core definitions that will be used throughout the paper; Section III introduces the features that support our proposed classification tool; Section IV describes the tool design; Section V presents the related work in the web page classification area; finally, Section VI lists some of the conclusions drawn from the research and concludes the article.

II. Core Definitions

Next, we introduce some necessary concepts that will be used throughout the rest of the paper.

Hubset We define a hubset H as a set of hubs obtained from a particular site.

$$H = \{H_1, H_2, H_3, ..., H_n\}, n \geq 1 \tag{1}$$

We define a hub H_i as the set of links l_j it contains.

$$H_i = \{l_1, l_2, l_3, ..., l_m\}, m \geq 1 \tag{2}$$

Linkset For each hubset H, we define its linkset L as the set of links contained in H, that is, the set of links from every H_i in H.

$$L = \bigcup_{i=1}^{n} H_i \in H, n \geq 1 \tag{3}$$

Link We define a link l as a tuple that represents a URL, and is composed of

$$l = (S, A, P, N, V) \tag{4}$$

Where S is the schema of the URL, A is its authority or domain name, P is a sequence of path segments, N is a sequence of names of the parameters in the URL query string and V is the sequence of the former parameters values.

Links are obtained from URLs by means of a tokeniser, according to RFC 3986. Figure 2 show an schema of link tokenisation, using URL http://scholar.google.com/scholar?q=Web+crawling as an example. Note that RFC 3986 states that links also include an optional fragment preceded by a '#' symbol, which points to a specifical section inside a page. However, our goal is to build URL prototypes of URL that link to pages about a particular concept, and not specifical sections inside pages, therefore the fragment part of an URL is not useful for our purposes. Throughout this paper we will omit the fragment, although including it in the process will not affect the results.

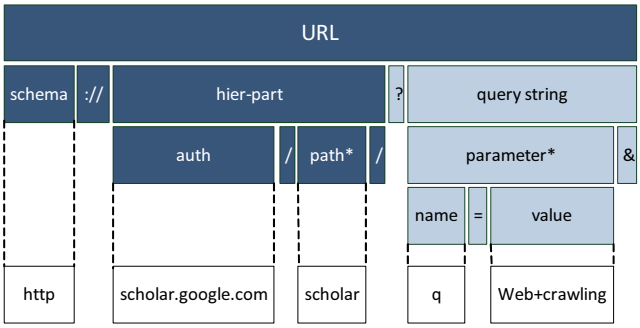


Fig. 2: RFC 3986 simplified URL schema with as an example of tokenisation

Prototype We define a prototype p as a link

$$p = (S, A, P, N, V) \tag{5}$$

where each element in P, N and V is either a literal or a wildcard, $\star$. In this case, a wildcard represents any sequence of characters (excluding separators '?', '/', '#', '=' and '&').

Common Path Links Let L be a linkset from a given site, $l = (S, A, P, N, V)$ be a link in L and $P(i)$ be the i-th path element of l, $P(i) \in P$. We define the set CPL as the set of all links l' in L having the same prefix as l up to (and excluding) $P(i)$, being $l' = \{S', A', P', N', V'\}$.

$$CPL_P(l, i) = \{l' \in L \cdot (S, A) = (S', A') \wedge P(j) = P'(j), j \in [1, i)\} \tag{6}$$

Similarly, we define CPL for the parameter names and values in l as the set of links l' in L having the same prefix as l ip to (and excluding) $N(i)$ or $V(i)$.

$$CPL_N(l, i) = \{l' \in L \cdot (S, A, P) = (S', A', P') \wedge N(j) = N'(j) \wedge V(k) = V'(k), j \in [1, i), k \in [1, i-1)\} \tag{7}$$

$$CPL_V(l, i) = \{l' \in L \cdot (S, A, P) = (S', A', P') \wedge N(j) = N'(j) \wedge V(k) = V'(k), j, k \in [1, i)\} \tag{8}$$

Recall that a prototype is nothing more than a link that includes some wildcard sections, so we can calculate the CPL (p,i) set of a prototype likewise.

Example As an example, consider p defined in Figure 3a. Prototype p represents all links starting with http://www.amazon.com/, followed by any combination of characters except separators, then the literal string /dp?ie=UTF-8&qid=, and another combination of characters at the end. Represented links are depicted in Figure 3c. If we check links in Amazon, we find out that links following this pattern are links to product detail pages.

Fig. 1: Example of Link Classification: Amazon.com hub page.

Let L be a linkset composed of links in Amazon, as shown in Figure 3b, and l_1 be the first link in Figure 3b. The $CPL_P(l_1, 1)$ set is shown in Figure 3d.

III. Classification Features

Next, we introduce the features that support building the set of prototypes that represent all links in a given site. We take a statistical approach to the problem of prototype building, and we base our technique in the definition of probabilistic features for each link and each token inside a link. First we give a formal definition of these features and later we illustrate their use by means of an example.

A. Features Definition

Link feature Let H be a hubset from a certain web site with size n, and l be a link $l \in H$. Probability F_L of a link in the context of H is defined as follows.

$$F_L(l) = \frac{|\{H_i \in H \cdot l \in H_i, i \in [1, n]\}|}{n} \quad (9)$$

In Equation 9, we must assure that the hubset is sufficiently large so that the probability estimation is statistically significant, so we require that $|H| \geq 30$, which is the usual threshold in statistical literature. F_L (l) takes values in the range $[1/n, 1]$. Links that appear more frequently in hubs from a hubset, have a higher F_L than those appearing just in a few of them, to the point that links with $F_L = 1$ appear in every single $H_i \in H$. At the other end of the distribution, links with F_L near to 0 never appear in any of H hubs.

As an example, Figure 4a shows the histogram of F_L values obtained from the 100 hubs in an e-commerce site (Amazon.com) an two academic sites (Microsoft Academic Search and TDG Scholar).

Tokens Features Let H be a hubset from a given site with size n, L its linkset, l a link of the form (S, A, P, N, V) in L and $X(i)$ be the i-th element of X, we define the feature value of $X(i)$ given as the following probability.

$$F_X(l, i) = \frac{|\{H_j \in H \cdot H_j \cap CPL_X(l, i+1) \neq \emptyset, j \in [1, n]\}|}{n} \quad (10)$$

These features values are in the same range as F_L, $[1/n, 1]$. Same as with F_L, path segments that appear more frequently in hubs from H have a higher F_P than those that only appears in URLs from some of the hubs.

Figure 4b shows the histogram of F_P, F_N and F_V values from the same hubsets and sites as defined for F_L values. It is noticeably similar to the F_L histogram presented earlier, with the majority of values around $1/n$, and just a small tail near 1.

Given that a prototype has the same signature as a link, both previous definitions III-A and III-A are applicable as well to prototypes. For the sake of simplicity, we assume that $F_P(p, i) = 1 \iff p = \{S, A, P, N, V\} \wedge P(i) = \star$ (similarly, with F_N and F_V).

B. Features Examples

Example Consider an experiment over Amazon.com, in which we issue 100 queries using the top 100 words in English language, discarding stop words. The result of this experiment is a hubset H composed of $n = 100$ hubs.

The F_L values calculated for some of the links in H are shown in Table I

All Amazon pages contain a navigation bar in the upper part of the page, including useful links such as “Home” “Sign In”

http	www.amazon.com	★	dp	ie	qid	UTF-8	★

a) Prototype p

LINK	S	A	P			N		V	
			1	2	3	1	2	1	2
l_1	http	www.amazon.com	Head-First-Java	dp	∅	ie	qid	UTF-8	130
l_2	http	www.amazon.com	Effective-Java	dp	∅	ie	qid	UTF-8	333
l_3	http	www.amazon.com	gp	site-directory	ref=topnav_sa	∅	∅	∅	∅
l_4	http	www.amazon.com	Head-First-Java	p-r	∅	ie	qid	UTF-8	130
l_5	http	www.amazon.com	Effective-Java	p-r	∅	ie	qid	UTF-8	333
l_6	http	www.amazon.com	Beginning-Programming	dp	∅	ie	qid	UTF-8	234
l_7	http	www.amazon.com	Sam-Teach-Java	dp	∅	ie	qid	UTF-8	123
l_8	http	www.amazon.com	Introduction-Java	dp	∅	ie	qid	UTF-8	130
l_6	http	www.amazon.com	Beginning-Programming	p-r	∅	ie	qid	UTF-8	234
...									
l_n	http	www.amazon.com	ref=gno_logo	∅	∅	∅	∅	∅	∅

b) Amazon Linkset

LINK	S	A	P		N		V	
			1	2	1	2	1	2
l_1	http	www.amazon.com	Head-First-Java	dp	ie	qid	UTF-8	130
l_2	http	www.amazon.com	Effective-Java	dp	ie	qid	UTF-8	333
l_3	http	www.amazon.com	Sam-Teach-Java	dp	ie	qid	UTF-8	123
l_4	http	www.amazon.com	Beginning-Programming	dp	ie	qid	UTF-8	234
...								
l_n	http	www.amazon.com	Introduction-Java	dp	ie	qid	UTF-8	234

c) Links in L matching prototype p

LINK	S	A	P		N		V	
			1	2	1	2	1	2
l_1	http	www.amazon.com	Head-First-Java	dp	ie	qid	UTF-8	130
l_2	http	www.amazon.com	Head-First-Java	p-r	ie	qid	UTF-8	130

d) $CPL_P(l_1,2)$

Fig. 3: Example of Prototype

and "Help". Examples of these links URLs are, respectively, links with ID 2, 3 and 4, and they are always present in almost any page we choose from the site. Therefore, for any hubset extracted from Amazon, the probability of these URLs is always 1 or near 1.

On the other side, there are links whose appearance depends on the specifical page being considered. For example, links to a page with detailed information about a product, just like links with ID 1, 5 and 6 in the example, only appear in hubs which are responses to certain queries. Therefore, its probability depends on the hubset, although we can assume that, for a random set of hubs, F_L value is rather low.

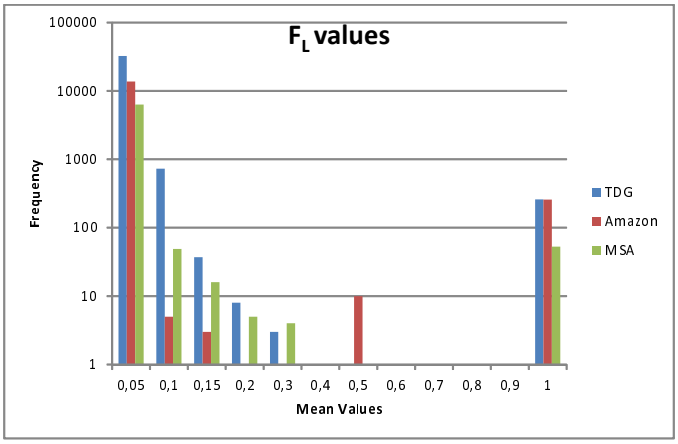
In general, our hypothesis is that for links whose F_L in a hubset H is not 1 (or near 1), it is in fact around $1/n$, that is, probability values are grouped around the two extremes of the distribution (0 and 1), and the number of links whose probability is in the middle of the distribution is very low. Back to Figure 4a, we observe that most values are grouped around 0.05, which means that most links just appear in a range of 1 to 5 hubs, approximately. We must note that there

ID	l	$F_L(l)$
1	http://www.amazon.com/Head-First-Java/dp?ie=UTF8&qid=130	0.01
2	http://www.amazon.com/ref=gno_logo	0.99
3	http://www.amazon.com/Help/b/ref=topnav_help?ie=UTF8&node=508510	0.99
4	http://www.amazon.com/gp/yourstore/ref=pd_irl_gw?ie=UTF8&signIn=1	1.00
5	http://www.amazon.com/Effective-Java/dp?ie=UTF8&qid=130	0.01
6	http://www.amazon.com/Head-First-Java/product-reviews?ie=UTF8	0.03

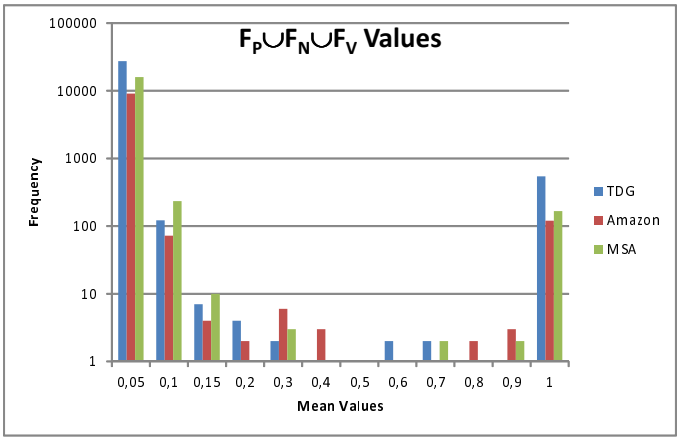
TABLE I: Values for feature F_L in Example III-B

is a small but significant group of values around 1, that is, the group of links that are present in every hub from the site. We can therefore conclude that links with $F_L = 1$ are those belonging to the site template. Hence, our technique allows us to detect the template of a given site, besides classifying its links according the concept contained in their targets.

Let l_1 be the link with ID = 1 in H defined previously. After the experiment, we obtained the values for features F_P, F_N and F_V presented in Table II. As a comparison, in Table III we show the values for features F_P, F_N and F_V for the



(a) F_L values histogram



(b) F_P, F_N, F_V values histogram

Fig. 4: F_P, F_N, F_V and F_L values histogram, from sites: Amazon.com, TDG Scholar and Microsoft Academic Search

l_1 = http://www.amazon.com/Head-First-Java/dp?ie=UTF-8&qid=123		
	i = 1	i = 2
$F_P(l_1, i)$	0.01	0.01
$F_N(l_1, i)$	0.01	0.01
$F_V(l_1, i)$	0.01	0.01

TABLE II: Values for feature F_P, F_N and F_V for link l_1 in Example III-B

prototype p that results when we replace the first path segment in l ("dp") with a wildcard. Based on the former example, we can extract some conclusions from the different values of F_P, F_N and F_V. For example, token "dp", with $F_P(p, 2) = 0.98$, is a fixed part of every link to Amazon product detail pages, and therefore, it is more frequent throughout the site than token "123", whose $F_V(p, 2)$ is near 0 as it is a parameter that identifies queries, and therefore, it is different for every issued query. As a result, its F_V value is 0.01, indicating that it just appears in links from a single hub. Similarly, parameter 1, with name "ie" and value "UTF-8", is also a fixed part in all Amazon links, so their F_N and F_V values respectively are

p = http://www.amazon.com/★/dp?ie=UTF-8&qid=123		
	i = 1	i = 2
$F_P(p, i)$	1	0.98
$F_N(p, i)$	0.99	0.99
$F_V(p, i)$	0.99	0.01

TABLE III: Values for feature F_P, F_N and F_V for prototype p in Example III-B

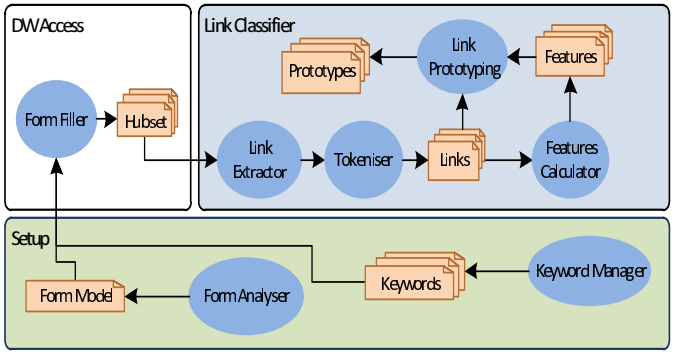


Fig. 5: Diagram of the architecture.

near to 1 in Table III. Our hypothesis regarding F_P, F_N and F_V values is the same exposed earlier for F_L values. In this case, the straightforward application is to build prototypes: tokens with a near-zero value are not relevant, so we can abstract over them and obtain a more general representation of all such segments in the form of a regular expression, that is, of a prototyping token. Meanwhile, tokens with a feature value significantly higher than the others (usually around 1) appear in most hubs, so they are part of the characteristic URL patterns used the site to compose URLs. In other words, they are relevant, and they are not grouped with others to form a prototype; instead, they stay as literals.

IV. Classification Tool

Based on the previous features, we implemented a link classifier, following the architecture in Figure 5. First, a training set is needed, composed by links from the site we wish to extract information from. For this purpose, we make use of the Form Analyser which analyses the forms to obtain a form model, and the Form Filler that uses this model to automatically fill in the form and retrieve the resulting hubs, composing a hubset. Our proposal is focused on keyword-based queries, hence the form filler only deals with forms that contain at least one text field. A Keyword Manager is responsible for finding a corpus of keywords to be used by the form filler, trying to obtain the maximum number of hubs as possible, minimising the keywords that yield no result.

Afterwards, all URLs from the retrieved hubset are extracted and tokenised. For each link, values of features F_P, F_N, F_V and F_L, as defined in section III are calculated, and used to build an ordered set of prototypes, where each prototype represents a different class of links, that is, links leading to pages containing a different concept. Empirical results show

that it is indeed necessary that prototypes keep an order, as we usually obtain some prototypes that subsume other prototypes, that is, a regular expression that is more general than other, and that matches all links matched as well by the latter. To avoid misclassifications, in cases like that we always give more priority to the most specifical prototype than to the most general one.

We developed a proof-of-concept application, obtaining promising evaluation values. Due to space limitations, we only include an example of the classification results in Figure 6, in which we observe Cluster 0 representing the site template links, Cluster 8 representing products, Cluster 9 representing product reviews and Cluster 12 representing authors, amongst others. A demo version of the application can be found in [16].

V. Related Work

A. Web page classification

Web page classification has been extensively researched, and several techniques have been applied with successful experimental results. In general, we catalogue classifiers according to the type and location of the classification features. There are three main trends in feature types: content-based, structure-based and hybrid classifiers. As for feature location, most approaches obtain features from the page to be classified, while others get them from neighbour pages.

Content-based classifiers ([17], [26]) categorize a web page according to the words and sentences it contains. This kind of classifiers group all pages within the same topic, assigning them the same class label. As for structure-based classifiers ([3], [4], [6], [12], [27] and [28]), the main feature used to classify pages is their physical organisation of contents, usually expressed in a tree-like data structure, like a DOM Tree. Also, there are hybrid approaches, [9] and [21], which take into account both content and structural features.

All the previous classifiers consider different kinds of features, but in most cases those features are extracted from the page to be classified, which requires downloading it previously. There are also classifiers that explore the possibility of classifying a web page by using features extracted from neighbour pages, instead of the page itself, being the neighbour of a page another page that has a link to the former, or, conversely, that is linked from it. All these proposals are content-based, and usually rely on features such as the link anchor text, the paragraph text surrounding the anchor [11], the headers preceding the anchor, the words in the URL address, or even a combination of these [15]. If the link is surrounded by a descriptive paragraph or the link itself contains descriptive words, it is possible to decide the page topic in advance of downloading it.

VI. Conclusions

Our proposal classifies pages according to their URL format without downloading them beforehand. Parting from an unlabelled set of links, a set of prototypes is built, each of one representing all links to pages containing a concept embodied in a particular web site. The resulting prototype set can be used by a crawler to improve its efficiency by selecting in each page only links leading to pages with concepts that are interesting for the user, reaching those pages while downloading the minimum number of irrelevant pages. Besides, our classifier is able to detect the template of a web site, that is, links that appear in every page in the site, and hence will most probably not lead to information related to that query.

Using features located on a page to classify it requires previous download, which results in wasted bandwidth and time. There are some proposals that classify pages according to the text surrounding the link in the referring page. This improves the crawlers efficiency, but it is not a general technique, given that not all links include in their surroundings words useful for classification. Our proposal classifies web pages depending on the link URL format, so it is not only efficient, but also generic and applicable in different domains. Besides, user supervision is kept to a minimum, given that the classifier is trained using an unlabelled set of links collected automatically.

These links are analysed and a set of prototypes is built, each of them representing all URLs that link to pages containing a different concept. Therefore users do not have to label large training sets, as it happens in supervised classifiers; instead they are only responsible for defining his or her interest, by picking the related prototypes. Note that users intervention is unavoidable, given that the relevancy criteria depends solely on them, but in our proposal it is kept to a minimum.

Traditional crawlers browse the whole sites, retrieving all pages, and spending a significant time and bandwidth while downloading them. Focused crawlers retrieve pages belonging to a topic more efficiently than traditional crawlers do, but still they are not a suitable solution for virtual integration systems, because a page has to be classified to know if the crawler must follow that path, and that requires the page to be downloaded in most cases.

The prototypes thus generated can later be used by a crawler to classify links following the template matching approach, that is, links are compared against prototypes, and assigned to the first class whose prototypes they match. If the link class is marked as relevant by the user, the links are followed by the crawlers; otherwise, they are ignored, making the crawler more efficient. As a result, we designed a link classifier that lays the foundations for an efficient crawler, able to access web pages automatically, while requiring as little intervention as possible from the users.

Acknowledgment

This paper was supported by the European Commission (FEDER), the Spanish and the Andalusian R&D&I programmes (grants TIN2007-64119, P07-TIC-2602, P08-TIC-4100, TIN2008-04718-E, TIN2010-21744, TIN2010-09809-E, TIN2010-10811-E, and TIN2010-09988-E).

Fig. 6: Example of Link Classification: Amazon.com hub page.

REFERENCES

[1] Charu C. Aggarwal, Fatima Al-Garawi, and Philip S. Yu. On the design of a learning crawler for topical resource discovery. *ACM Trans. Inf. Syst.*, 19(3):286–309, 2001.

[2] Vinod Anupam, Juliana Freire, Bharat Kumar, and Daniel F. Lieuwen. Automating web navigation with the webvcr. *Computer Networks*, 33(1-6):503–517, 2000.

[3] Arvind Arasu and Hector Garcia-Molina. Extracting structured data from web pages. In *SIGMOD Conference*, pages 337–348, 2003.

[4] Ziv Bar-Yossef and Sridhar Rajagopalan. Template detection via data mining and its applications. In *WWW*, pages 580–591, 2002.

[5] Claudio Bertoli, Valter Crescenzi, and Paolo Merialdo. Crawling programs for wrapper-based applications. In *Information Reuse and Integration*, pages 160–165, 2008.

[6] Lorenzo Blanco, Valter Crescenzi, and Paolo Merialdo. Structure and semantics of Data-IntensiveWeb pages: An experimental study on their relationships. *J. UCS*, 14(11):1877–1892, 2008.

[7] Lorenzo Blanco, Nilesh Dalvi, and Ashwin Machanavajjhala. Highly efficient algorithms for structural clustering of large websites. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 437–446, New York, NY, USA, 2011. ACM.

[8] Jim Blythe, Dipsy Kapoor, Craig A. Knoblock, Kristina Lerman, and Steven Minton. Information integration for the masses. *J. UCS*, 14(11):1811–1837, 2008.

[9] James Caverlee and Ling Liu. Qa-pagelet: Data preparation techniques for large-scale data analysis of the deep web. *IEEE Trans. Knowl. Data Eng.*, 17(9):1247–1262, 2005.

[10] Soumen Chakrabarti. Focused web crawling. In *Encyclopedia of Database Systems*, pages 1147–1155. 2009.

[11] William W. Cohen. Improving a page classifier with anchor extraction and link analysis. In *NIPS*, pages 1481–1488, 2002.

[12] Valter Crescenzi, Giansalvatore Mecca, and Paolo Merialdo. RoadRunner: Towards automatic data extraction from large web sites. In *Very Large Data Bases*, pages 109–118, 2001.

[13] Guilherme T. de Assis, Alberto H. F. Laender, Marcos André Gonçalves, and Altigran Soares da Silva. Exploiting genre in focused crawling. In *String Processing and Information Retrieval*, pages 62–73, 2007.

[14] Jenny Edwards, Kevin S. McCurley, and John A. Tomlin. An adaptive model for optimizing performance of an incremental web crawler. In *WWW*, pages 106–113, 2001.

[15] Johannes Fürnkranz. Hyperlink ensembles: a case study in hypertext classification. *Information Fusion*, 3(4):299–312, 2002.

[16] Inma Hernández. Relc demo. http://www.tdg-seville.info/inmahernandez/Thesis+Demo, 2011.

[17] Andreas Hotho, Alexander Maedche, and Steffen Staab. Ontology-based text document clustering. *KI*, 16(4):48–54, 2002.

[18] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.

[19] Juliano Palmieri Lage, Altigran Soares da Silva, Paulo Braz Golgher, and Alberto H. F. Laender. Automatic generation of agents for collecting hidden web pages for data extraction. *Data Knowl. Eng.*, 49(2):177–196, 2004.

[20] Stephen W. Liddle, David W. Embley, Del T. Scott, and Sai Ho Yau. Extracting data behind web forms. In *ER (Workshops)*, pages 402–413, 2002.

[21] Alex Markov, Mark Last, and Abraham Kandel. The hybrid representation model for web document classification. *Int. J. Intell. Syst.*, 23(6):654–679, 2008.

[22] Sougata Mukherjea. Discovering and analyzing world wide web collections. *Knowl. Inf. Syst.*, 6(2):230–241, 2004.

[23] Alberto Pan, Juan Raposo, Manuel Álvarez, Justo Hidalgo, and Ángel Viña. Semi-automatic wrapper generation for commercial web sources. In *Engineering Information Systems in the Internet Context*, pages 265–283, 2002.

[24] Gautam Pant and Padmini Srinivasan. Link contexts in classifier-guided topical crawlers. *IEEE Trans. Knowl. Data Eng.*, 18(1):107–122, 2006.

[25] Ioannis Partalas, Georgios Paliouras, and Ioannis P. Vlahavas. Reinforcement learning with classifier selection for focused crawling. In *European Conference on Artificial Intelligence*, pages 759–760, 2008.

[26] Ali Selamat and Sigeru Omatu. Web page feature selection and classification using neural networks. *Inf. Sci.*, 158:69–88, 2004.

[27] Márcio L. A. Vidal, Altigran Soares da Silva, Edleno Silva de Moura, and João M. B. Cavalcanti. Structure-based crawling in the hidden web. *J. UCS*, 14(11):1857–1876, 2008.

[28] Karane Vieira, Altigran Soares da Silva, Nick Pinto, Edleno Silva de Moura, João M. B. Cavalcanti, and Juliana Freire. A fast and robust method for web page template detection and removal. In *CIKM*, pages 258–267, 2006.

[29] Yang Wang and Thomas Hornung. Deep web navigation by example. In *BIS (Workshops)*, pages 131–140, 2008.

[30] Lan Yi, Bing Liu, and Xiaoli Li. Eliminating noisy information in web pages for data mining. In *Knowledge Discovery and Data Mining*, pages 296–305, 2003.

Unsupervised Texture Image Classification using Self-Organizing Maps

Nedyalko Petrov[1] and Ivan Jordanov[2]

School of Computing, University of Portsmouth,
Buckingham Building, Lion Terrace, Portsmouth, Hampshire, PO1 3HE, UK
[1]Nedyalko.Petrov@port.ac.uk
[2]Ivan.Jordanov@port.ac.uk

***Abstract* - In this paper we report results from unsupervised texture image classification on a data set of images collected with our intelligent machine vision system for pattern recognition (assuming no *a priori* human vision expert knowledge is available for the image classes). The simulation of the investigated system includes four main phases: data collection and feature extraction, feature analysis, classifier training, and classifier testing and evaluation. Self-Organizing Maps (SOM) are used for classification of the collection of images into several classes, based on their features and texture characteristics. Three main experiments are conducted during this research: in the first one, all extracted features are used for training the classifiers without any statistical pre-processing of the dataset; in the second simulation, the classifiers are trained after normalization of the available data; and in the last experiment, the trained SOMs use linear transformations of the original features, received after pre-processing with principal component analysis (PCA). Each test is performed 50 times and the classification results are assessed using three commonly applied metrics, namely: accuracy rate, sensitivity and specificity. Finally, the findings of this investigation are compared with results from other authors.**

***Keywords* – image analysis, feature extraction, texture classification, self-organizing maps, unsupervised learning, principal component analysis.**

I. Introduction

The analysis, recognition and classification of texture patterns and images are important topics in the field of digital image processing and pattern recognition, with many areas of applications [1], [2], [5], [13] and [11].

Investigated in the literature texture classification systems vary in their approaches, but most of them include data collection, data pre-processing, feature extraction, feature analysis, classification and evaluation stages [16], [17] and [18]. The initial processing and transformation of the image raw data is difficult but important part of the whole process, which aim is to identify appropriate data characteristics and features that are to be used in the later stages [16]. Often, the raw data is too large or complex to be used directly as input to a classifier, leading to the 'curse of dimensionality' and other training problems related to the generalization abilities of the learned systems, especially when insufficient training samples are available. Even if this is not the case, reducing the number of variables representing the data can speed up and facilitate the learning process at later stages [17]. That is why principal component analysis (PCA), for example, is a widely accepted technique in such cases [1], [2], and [6].

In [6] we investigated a supervised classification of texture images problem using supervised neural network learning for which *a priori* knowledge about the image classes was used. The aim of this research is to extend the previous investigation, considering the same classification problem, but assuming there is no expert knowledge available for the texture classes of the data set samples.

Unsupervised classification of texture patterns and images is widely used approach with applications in a broad range of areas, for example: in [1] it is used to determine water quality based on some chemical and physicochemical features; in [2] it is applied for classification of SAR images; in [3] for texture-based classification of atherosclerotic carotid plaque images for determining risk of stroke for individuals; in [5] surface texture features are used for classifying volcanic ash; in [10] a SOM is trained to automatically classify texture structure of different fabric types; in [12] the authors apply it for classification of textures in scene images using biology inspired features; in [15] the researchers employ SOMs for classification of aerial images, etc.

In this investigation we use a data set of 335 texture images, acquired with an intelligent visual recognition system, as reported in [6]. In the referenced work, each data sample of the set represents a greyscale image of an industrial cork tile that was classified in one of seven predefined classes – *Beach*, *Corkstone*, *Desert*, *Lisbon*, *Pebble*, *Precision* and *Speckled*. The distribution of the texture classes is non-uniform, as shown in Fig. 1.

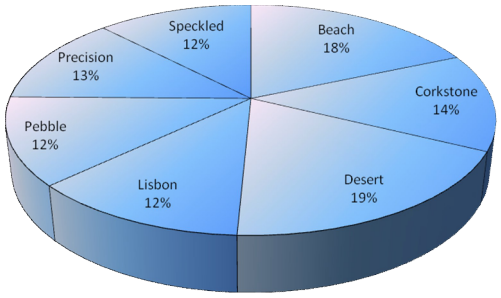


Fig. 1 Texture classes' distribution (non-uniform).

The simulation of the investigated system is divided in four main stages: data collection and feature extraction, feature

analysis, classifier training, and classifier testing and evaluation.

The rest of the paper is organized as follows: Section II presents information about the data acquisition and the feature extraction stage. The feature analysis is described in Section III and Section IV covers the classification phase. The results of the conducted tests are reported and discussed in Section V. Section VI concludes the paper and gives some ideas for future work.

II. Data Collection and Features Extraction

The texture image data collection stage is not of prime interest of this paper. The images used for testing the model are collected in [6] via an intelligent visual recognition system. The data set consists of 335 greyscale images of size 230x340 pixels of cork tile samples, each of which belongs to one of seven predefined by experts classes.

A set of characteristics (features), that represent some valuable information about the texture of the images, is obtained and three different techniques are used for extracting a total of 34 features. The first eight of them are received by calculating co-occurrence matrices for the images. This approach, proposed by Haralick in [7], is a commonly applied statistical method for feature extraction. It takes into account relative distances and orientation of the pixels with co-occurring values [8], [15] and [18]. As usually proposed by other authors [14], four relative orientations are used – horizontal (0º), right diagonal (45º), vertical (90º), and left diagonal (135º). In this way the *energy*, *homogeneity*, *correlation*, and *contrast* characteristics in each direction are computed [17], [18]. Also, two spatial relationships are considered – the direct neighbours and the pixels with difference of five. In this way, a total of eight co-occurrence matrices are obtained – four for the direct neighbours and another four for the pixels with difference of five.

Another 25 features are produced by applying Law's masks filter technique, which is used to identify points of high energy in an image [4]. As in [3] and [6], 1x5 pixel masks designed to compute the average *grey level*, *edges*, *ripples*, *spots* and *waves* are employed in this research. Each mask is multiplied by each other and this way 25 different masks are obtained.

The last feature represents entropy of a greyscale image. It is a statistical measure of randomness that can be used to characterize the image texture. It takes low values for smooth images and vice versa. The entropy for each image is calculated using a MATLAB's build-in function according to Equation 1:

$$E = -\sum_{i=0}^{255} p_i . \log_2 p_i \qquad (1)$$

where p_i contains the histogram counts for each level of grey (256 gray levels) in the image.

III. Feature Analysis and Reduction

Before applying any statistical analysis, a subset consisting of 25% of the available data for each class is excluded for the testing purposes. It is randomly selected and from now on will be referred to as testing set. The remaining 75% of the available data will be the training set.

The distribution of the seven classes of the training set, represented by two randomly selected from the extracted 34 features is shown on Fig. 2. The considerable overlapping between the classes makes the classification process a difficult task.

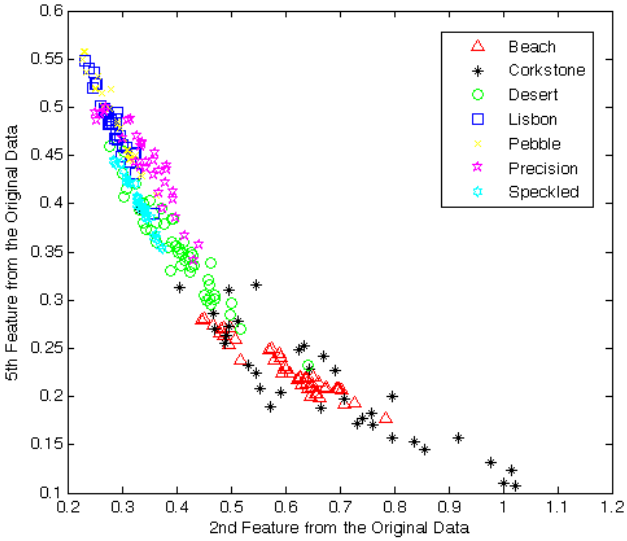


Fig. 2 Classes distribution (according to two randomly selected features from the training set) shows considerable overlapping between the classes.

In order to get rid of at least some of the redundant information (i.e., the information contained in highly correlated characteristics), and to reduce the dimensionality of the classification problem in hand (which is in straight correlation to the number of inputs to the neural network), Principal component analysis is used in some of the experiments. PCA is an unsupervised, eigenvalue based multivariate technique that transforms a number of possibly correlated variables into a smaller number of uncorrelated variables, called principal components (PC), without using any labelled information of the data [10], [16]. The first PC accounts for as much of the information variance in the data, as possible, and each succeeding PC accounts for as much of the remaining variability as possible.

Depending on the areas of application, PCA is also referred as Hotelling transform, Karhunen-Loeve transform (KLT), or proper orthogonal decomposition (POD), [18].

For the implementation of PCA, processing the available data and extracting training features as inputs for the learning process, MATLAB's Statistics Toolbox is used. As a result, a new data set is obtained, in which the first five principal components contain about 97% of the total information variance, as it is illustrated in Fig. 3. The PCA transformation matrix (which allows backward transformation to the original features), is of course saved for further use in the testing and evaluation stages.

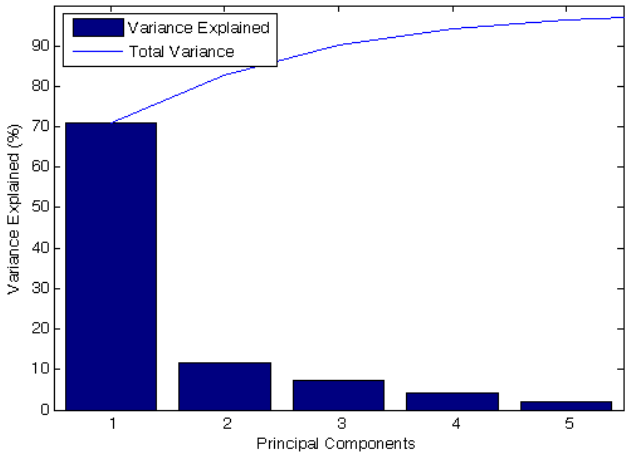


Fig. 3 Percentage of the information from the training set for the PCA experiment. The first five principle components contain about 97% of the total information variance.

Fig. 4 shows the distribution of the seven texture classes, represented by the first and second PCs. It can be seen that classes' separability is considerably improved (compared to fig. 2), and four out of the seven classes (*Beach*, *Corkstone*, *Desert* and *Pebble*) are more easily separable from the others, using the first two PCs only.

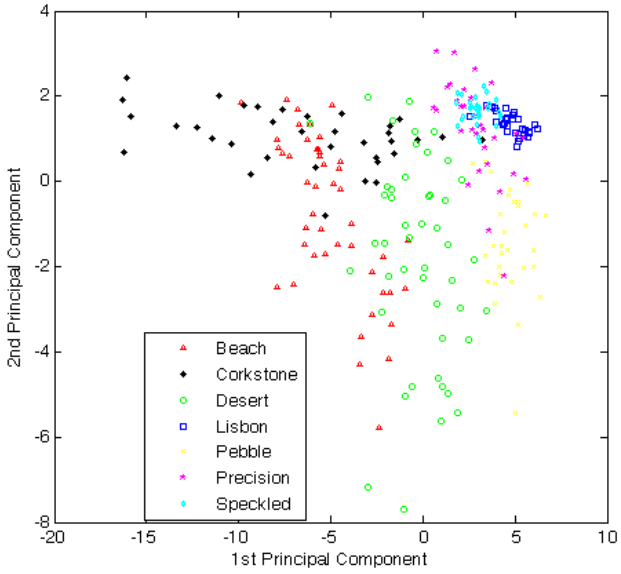


Fig. 4 Classes distribution, according to the first two PCs shows that the separability is considerably improved.

IV. Classification

In this stage, self-organising maps (SOM) are employed for classification of the texture samples data. A SOM is an artificial neural network that is trained using unsupervised learning to produce a low-dimensional (typically two-dimensional), discretized representation of the input space of the training samples, called map. SOMs differ from other neural networks in that they use neighbourhood function to preserve the topological properties of the input space [9]. Like most neural networks, SOMs operate in two modes: training and mapping.

The classification is performed according to the following algorithm:

1) SOM with specific architecture is created (number of neurons, map topology, learning parameters, etc.);
2) The training set (the extracted features for 75% of the available data) is presented to the SOM for training.
3) As a result of the training a 2D map is obtained, in which each node and its closest neighbours represent similar data samples (Fig. 5);
4) The count of the samples belonging to a certain class is determined for each node of the map, based on the available expert knowledge for the training samples;
5) Each node is labelled to the predominant class presented in it. In case equal number of samples of different classes is mapped to a certain node, the node is labelled to the predominant class in its neighbourhood (Fig. 5). A node gets no label if there are no data samples mapped to it (the red node on Fig. 5b);

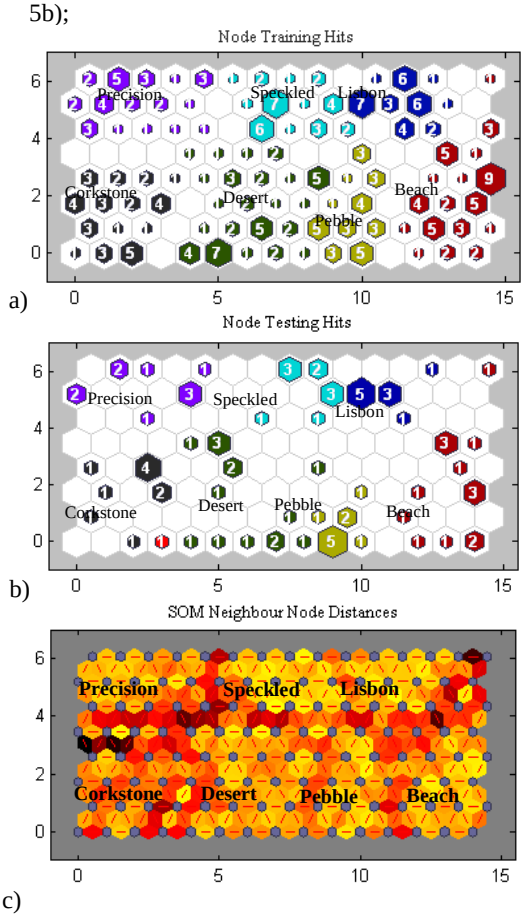


Fig. 5 Sample SOM classifier map. Image a) presents the node hits for the samples from the training set and b) from the testing set. The number of hits are given in each node. The nodes are coloured according to the classes they are labelled to. Image c) shows the relative distance between the map nodes. The darker colour corresponds to larger distance.

6) The classifier is tested with the remaining 25% of the available data;
7) Each testing sample label is compared to the label of the node that it is mapped to. A sample is counted as unclassified if it is mapped to an unlabelled node.

V. Evaluation and Results

MATLAB 2010B and its Neural Network, Image Processing and Statistics Toolboxes are used for performing all computations and simulations presented in this paper.

Three major experiments are conducted. In the first one, classifiers are trained using all the extracted features without any statistical pre-processing. In the second, the extracted features are normalized before being fed to a classifier and in the last one, the trained classifiers use features obtained after pre-processing with PCA.

The results are assessed using three metrics:

1) Accuracy rate – the classification accuracy rate is calculated according to Equation 2:

$$a = \frac{n_c}{n_c + n_w + n_u} \cdot 100\ [\%] \qquad (2)$$

where a is the accuracy of the classifier, n_c is the number of correctly classified samples, n_w is the number of wrongly classified samples and n_u is the number of unclassified samples.

2) Sensitivity – this metric measures the proportion of testing samples belonging to a certain class that are correctly classified to it. It is calculated for each class (C) by Equation 3:

$$Sensitivity_C = \frac{Correctly\ classified\ samples\ in\ C}{Total number of samples\ classified as\ C} \qquad (3)$$

3) Specificity – shows the proportion of testing samples not classified to a certain class (i.e., classified to any other class) to the all testing samples not belonging to that class. It is calculated for each class (C) by Equation 4:

$$Specificity_C = \frac{Samples\ correctly\ not\ classified\ in\ C}{Total\ number\ of\ samples\ not\ classified\ in\ C} \qquad (4)$$

During the simulation, each test is performed 50 times using the algorithm given in Section IV. The minimum, maximum and mean percentages of successfully classified texture images from the testing set are recorded and the mean standard deviation over the 50 runs is also calculated.

A. Classification without Statistical Pre-processing

As expected from the title, in this classification simulation, all the 34 extracted features are not subjected to statistical pre-processing and random 75% (251 texture images) of the available data samples are used for training the SOMs and subsequently, the remaining 25% (84 texture images) are used for testing.

Simulations with varying number of training epochs, number of neurons in the SOM topology, and the SOM's architecture are also conducted. The obtained results are presented in Table 1 and Table 2 and the sample confusion matrix is given in Table 3. The sensitivity and specificity metrics presented in Table 10 show that this classifier performs better for some of the classes (e.g., *Beach, Lisbon* and *Speckled*), but not for all of them.

Table 1 Variation of the classifier's accuracy (in %) for different number of training epochs and no statistical pre-processing. The table presents results over 50 runs for SOMs with 120 neurons and 15x8 map topology.

Epochs	**50**	**100**	**250**	**500**	**1000**	**2500**	**5000**	**7500**
Min	48.15	58.02	70.34	70.37	75.31	75.31	74.07	75.31
Max	62.96	75.31	81.48	80.25	81.48	81.48	82.72	82.72
Mean	55.14	66.74	77.04	77.04	78.40	78.33	78.00	78.12
Std	3.62	3.91	2.56	1.88	1.42	1.58	1.89	1.79

Table 2 Variation of the classifier's accuracy (in %) for different number of neurons, different SOM topology and no statistical pre-processing. The table presents results over 50 runs for SOMs trained for 500 epochs.

Number of Neurons	**60**			**120**		
SOM Topology	**3x20**	**5x12**	**6x10**	**6x20**	**10x12**	**12x10**
Min	70.37	69.14	69.14	67.90	70.37	70.37
Max	82.72	79.01	80.25	81.48	81.48	81.48
Mean	77.93	75.16	75.09	75.51	75.85	76.62
Std	2.49	2.37	2.32	2.86	2.00	2.45

Table 3 Sample confusion matrix for SOM classifier with 120 neurons (15x8 map topology) and 500 training epochs without statistical pre-processing.

Actual \ Predicted	***Beach***	***Corkstone***	***Desert***	***Lisbon***	***Pebble***	***Precision***	***Speckled***	***NC***
Beach	**14**	1	0	0	0	0	0	0
Corkstone	1	**8**	0	0	1	0	0	1
Desert	2	0	**10**	0	1	1	1	0
Lisbon	0	0	0	**11**	0	0	0	0
Pebble	0	1	0	2	**8**	0	0	0
Precision	1	0	1	1	2	**5**	1	0
Speckled	0	0	0	1	0	0	**9**	0

B. Classification with feature normalization

In this case, the training set is normalized so that the features have zero mean and unity standard deviation. The SOMs are trained using all 34 features and the learning process include experiments with varying number of training epochs, changed number of SOM's neurons, and different SOM's topologies.

Table 4 Variation of the classifier's accuracy (in %) for different number of training epochs for SOM with 120 neurons (15x8 map topology) after normalization. The table presents results over 50 runs.

Epochs	**50**	**100**	**250**	**500**	**1000**	**2500**	**5000**	**7500**
Min	71.60	79.01	83.95	83.95	85.19	85.19	87.65	87.65
Max	86.42	90.12	93.83	93.83	93.83	93.83	95.06	93.83
Mean	77.83	84.86	88.71	89.75	89.85	89.80	90.79	90.86
Std	3.55	3.14	2.43	1.98	2.05	1.80	1.82	1.56

The results are given in Table 4 and Table 5, and the sample confusion matrix is presented in Table 6. The sensitivity and specificity metrics, describing the accuracy of the neural networks, are given in Table 10. Their analysis show that the classifier's performance is improved, demonstrating better ability to distinguish between most of the classes, but still experiencing some difficulties with the *Corkstone* and the *Pebble* samples.

Table 5 Variation of the classifier's accuracy (in %) for different number of neurons and different SOM topology after normalization. The table presents results over 50 runs for SOMs trained for 500 epochs.

Number of Neurons	60			120		
SOM Topology	3x20	5x12	6x10	6x20	10x12	12x10
Min	82.71	83.95	85.19	83.95	83.95	85.19
Max	92.59	92.59	92.59	93.83	92.59	92.59
Mean	87.95	88.07	87.90	88.10	89.04	89.06
Std	1.99	2.11	1.948	2.04	1.86	1.64

Table 6 Sample confusion matrix for SOM classifier with 120 neurons (15x8 map topology) and 500 training epochs after normalization

Predicted / Actual	*Beach*	*Corkstone*	*Desert*	*Lisbon*	*Pebble*	*Precision*	*Speckled*	*NC*
Beach	**13**	1	0	0	0	0	0	1
Corkstone	0	**7**	1	0	2	0	0	1
Desert	0	0	**15**	0	0	0	0	0
Lisbon	0	0	0	**11**	0	0	0	0
Pebble	0	0	1	0	**10**	0	0	0
Precision	0	0	0	1	0	**10**	0	0
Speckled	0	0	0	0	1	0	**9**	0

C. Classification with PCA

Again, random 75% (251 texture images) of the available data samples are used for training and the remaining 25% (84 texture images) for testing, the difference being that the data used for SOMs training is statistically pre-processed with PCA. Similarly to the previous case, the number of training epochs, the number of neurons, the SOM's topology, and the number of PCs used for the training are varied. Each sub-experiment is performed 50 times and the minimal, maximal, and the mean accuracy for these runs are recorded. The results are given in Table 7 and Table 8, and illustrated in Fig. 6. The sample confusion matrix (Table 9) and the sensitivity and specificity metrics (Table 10) show that the classifier is having difficulties recognising some of the *Corkstone* samples, but its overall performance is very good, having over 90% accuracy for the other classes.

Table 7 Variation of the SOM accuracy (in %) for different number of training epochs (120 neurons, 15x8 map topology), and PCA pre-processing. Results over 50 runs for SOMs trained with the first 5 PCs

Epochs	50	100	250	500	1000	2500	5000	7500
Min	70.37	74.07	85.19	85.19	85.19	83.95	85.19	86.42
Max	85.19	88.89	92.59	91.36	93.83	92.59	92.59	92.59
Mean	75.60	80.94	89.06	88.79	89.23	88.86	89.46	89.28
Std	2.84	3.32	2.06	1.59	2.00	1.84	1.52	1.54

Table 8 Variation of the classifier's accuracy (in %) for different number of neurons and different SOM topology after PCA. The table presents results over 50 runs for SOMs trained with 500 epochs for the first 5 PCs.

Number of Neurons	60			120		
SOM Topology	3x20	5x12	6x10	6x20	10x12	12x10
Min	81.48	81.48	82.71	81.48	82.71	83.95
Max	91.36	92.59	91.36	93.83	92.59	91.36
Mean	86.74	87.78	87.43	87.11	88.74	88.35
Std	2.14	2.16	1.84	2.19	2.01	1.69

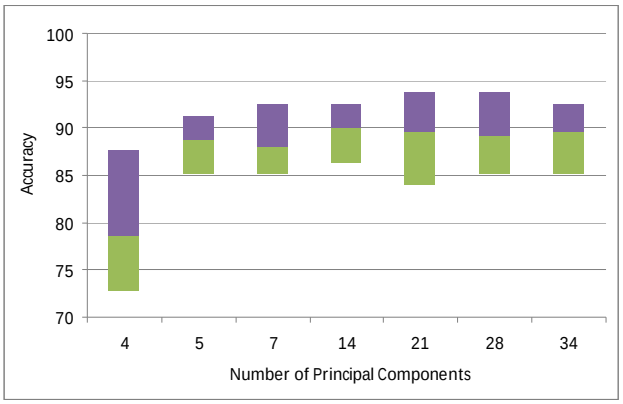


Fig. 6 Variation of the accuracy (in %) of the classifier (SOM with 120 neurons, 15x8 map topology, 500 epochs) for different number of PCs used during the training. The border between the sub-bars shows the mean accuracy rate for the 50 runs. The green and the purple sections show the min and max rate respectively.

Table 9 Sample confusion matrix for SOM classifier with 120 neurons (15x8 map topology) and 500 training epochs after PCA.

Predicted / Actual	*Beach*	*Corkstone*	*Desert*	*Lisbon*	*Pebble*	*Precision*	*Speckled*	*NC*
Beach	**14**	0	1	0	0	0	0	0
Corkstone	0	**7**	1	0	2	0	0	1
Desert	1	0	**14**	0	0	0	0	0
Lisbon	0	0	0	**11**	0	0	0	0
Pebble	0	0	0	0	**11**	0	0	0
Precision	0	0	0	1	0	**10**	0	0
Speckled	0	0	0	1	0	0	**9**	0

Table 10 Sample accuracy, sensitivity and specificity (in %) for SOM classifier with 120 neurons (15x8 map topology) and 500 training epochs.

	SOM			Normalized SOM			PCA		
	Accur	Sensit	Spec	Accur	Sensit	Spec	Accur	Sensit	Spec
Beach	93.3	77.8	98.5	86.7	100	97.2	93.3	93.3	98.5
Corkstone	72.7	80.0	95.9	63.6	87.5	94.7	63.6	100	94.8
Desert	66.7	90.9	93.2	100	88.2	100	93.3	87.5	98.5
Lisbon	100	73.3	100	100	91.7	100	100	84.6	100
Pebble	72.7	66.7	95.8	90.9	76.9	98.6	100	84.6	100
Precision	45.4	83.3	92.3	90.9	100	98.6	90.9	100	98.6
Speckled	90.0	81.8	98.6	90.0	100	98.7	90.0	100	98.7

D. Analysis of the results

The obtained results from the three case studies investigated in this work are presented in **Error! Reference source not found.**. It can be seen that, as expected, the worst accuracy is attained for the first case, where no statistical pre-processing of the extracted features is performed. Although the accuracy of the normalized data case (the second experiment) looks better than the one obtained for the PCA case, it has to be noted that only 5 PCs are considered during the training, whereas in the normalized case all 34 extracted features are taken into account. That is why the computational time in the last experiment is greatly reduced, compared to the other two.

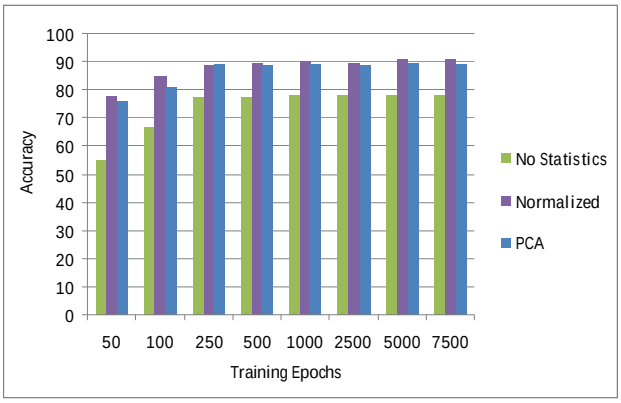


Fig. 7 Systems accuracy for different number of training epochs for the three case studies. As expected, lack of preprocessing of the training data leads to worst accuracy.

For the PCA experiment, the results shown in Fig. 6 illustrate that the mean system accuracy does not increase significantly when more that 5 PCs are considered for training. This is also supported by the plot given in Fig. 3, from which it can be seen that the first 5 PCs contain about 97% of the total data variation (information).

In regards to the SOM's topology, no clear correlation between the accuracy and the number of used neurons was observed (Table 2, Table 5 and Table 8). However, more experiments need to be conducted in order to investigate this issue in more detail.

Comparing the sample confusion matrices and sensitivity and specificity metrics for the three cases (Table 3, Table 6, Table 9 and Table 10), it can be said that the accuracy improved (as expected) after applying normalization and PCA on the extracted features, and this is especially valid for the *Desert*, *Pebble* and *Precision* classes, while at the same time the SOM kept excellent recognition rate for the *Beach*, *Lisbon* and *Speckled* classes.

It can also be seen from Table 1, Table 4 and Table 7 that increasing the number of epochs (above 250) for the runs does not lead to substantial improvement and the accuracy enters a plateau.

The obtained results from training after processing the original datasets with PCA (Table 7 and Table 8), show good agreement with the results of [2], where the authors reported between 81% and 98% accuracy for a PCA-based unsupervised classification of SAR images. Similar interval is achieved in [15] – (83%, 95.5%), and in [11] and [12] the investigated methods led to results with slightly larger accuracy variance, falling within (77%, 100%) and (67%, 92%) domains respectively.

VI. Conclusions

The investigated unsupervised approach for texture image recognition and classification problem includes statistical feature pre-processing techniques before the implementation of SOMs as classifiers. For the purpose of comparison, the experiments and simulations of the system are also conducted using the original data set without any statistical pre-processing. As expected, best results are obtained when statistical pre-processing (normalization) and PCA are implemented (Table 10).

The comparison of the sample confusion matrices for the three experiments (Table 3, Table 6 and Table 9), shows that the SOM classifiers generally confirm the experts' knowledge about the seven types of texture. However, the visual closeness of some of the misclassified samples to samples from other classes could assist experts to refine the classes' boundaries or to introduce new classes.

As discussed in the above paragraph, when our results are compared with results from other researcher reported in the literature, they show competitive and in some of the cases superior nature in terms of mean accuracy and its sensitivity and specificity derivative metrics.

References

[1] Astel A, Tsakouski S, Barbieri S, Simeonov V (2007) Comparison of self-organizing maps classification approach with cluster and principal components analysis for large environmental data sets. Water Research, 41, pp. 4566-4578.
[2] Chamundeeswari VV, Singh D, Singh K (2009) An Analysis of Texture Measures in PCA-Based Unsupervised Classification of SAR Images. IEEE Geoscience and Remote Sensing Letters, 6, pp. 214-218.
[3] Christodoulou CI, Pattichis CS, Pantziaris M, Nicolaides A (2003) Texture-based classification of atherosclerotic carotid plaques. IEEE Transactions on Medical Imaging, 22, pp. 902-912.
[4] Davies ER (2005) Machine vision : theory, algorithms, practicalities. Morgan Kaufmann, Amsterdam ; Oxford.
[5] Ersoy O, Aydar E, Gourgaud A, Artuner H, Bayhan H (2007) Clustering of volcanic ash arising from different fragmentation mechanisms using Kohonen self-organizing maps. Computers & Geosciences, 33, pp. 821-828.
[6] Georgieva A, Jordanov I (2009) Intelligent Visual Recognition and Classification of Cork Tiles With Neural Networks. IEEE Transactions on Neural Networks, 20, pp. 675-685.
[7] Haralick R, Shanmuga K, Dinstein I (1973) Textural Features for Image Classification. IEEE Transactions on Systems Man and Cybernetics, Smc3, pp. 610-621.
[8] Kohonen O, Hauta-Kasari M, Parkkinen J, Jaaskelainen T (2006) Co-occurrence matrix and Self-Organizing Map based query from spectral image database. art. no. 603305, ICO20: Illumination, Radiation, and Color Technologies, 6033, pp. 3305-3305.
[9] Kohonen T (1990) The Self-Organizing Map. Proceedings of the IEEE, 78, pp. 1464-1480.

[10] Kuo CFJ, Kao CY (2007) Self-organizing map network for automatically recognizing color texture fabric nature. Fibers and Polymers, 8, pp. 174-180.
[11] Lei Q, Zheng QF, Jiang SQ, Huang QG, Gao W (2008) Unsupervised texture classification: Automatically discover and classify texture patterns. Image and Vision Computing, 26, pp. 647-656.
[12] Martens G, Poppe C, Lambert P, Van de Walle R (2008) Unsupervised Texture Segmentation and Labeling Using Biologically Inspired Features. IEEE 10th Workshop on Multimedia Signal Processing, Vols 1 and 2, pp. 163-168.
[13] Paniagua B, Vega-Rodriguez MA, Gomez-Pulido JA, Sanchez-Perez JM (2010) Improving the industrial classification of cork stoppers by using image processing and Neuro-Fuzzy computing. Journal of Intelligent Manufacturing, 21, pp. 745-760.
[14] Randen T, Husoy JH (1999) Filtering for texture classification: A comparative study. IEEE Transactions on Pattern Analysis and Machine Intelligence, 21, pp. 291-310.
[15] Salah M, Trinder J, Shaker A (2009) Evaluation of the Self-Organizing Map Classifier for Building Detection from Lidar Data and Multispectral Aerial Images. Journal of Spatial Science, 54, pp. 15-34.
[16] Shih FY (2010) Image processing and pattern recognition: fundamentals and techniques. Wiley ; Chichester : John Wiley, Hoboken, N.J.
[17] Theodoridis S, Koutroumbas K (2009) Pattern recognition. Elsevier/Academic Press, Amsterdam; London.
[18] Umbaugh SE (2010) Digital image processing and analysis. CRC ; Taylor & Francis, Boca Raton, London.

A Transducer Model for Web Information Extraction

Hassan A. Sleiman, Inma Hernández, Gretel Fernández, Rafael Corchuelo
University of Seville
Departamento LSI, ETSI Informática
Avd. Reina Mercedes S/N, Sevilla 41012, Spain
Email: {hassansleiman,inmahernandez,gretel,corchu}@us.es

***Abstract*—In recent years, many authors have paid attention to web information extractors. They usually build on an algorithm that interprets extraction rules that are inferred from examples. Several rule learning techniques are based on transducers, but none of them proposed a transducer generic model for web information extraction. In this paper, we propose a new transducer model that is specifically tailored to web information extraction. The model has proven quite flexible since we have adapted three techniques in the literature to infer state transitions, and the results prove that it can achieve high precision and recall rates.**

***Index Terms*—Web Information Extraction, Inferring Transducers.**

I. Introduction

Web information provides added value in many automated business processes. Unfortunately, there are many web sites that do not provide a programmatic interface, which implies that such business processes need to extract and structure the information they require from HTML pages. During the last decades, this has motivated many researchers to work on information extractors [2]. A few rely on heuristics to identify repetitive data [1][11], but most of them are algorithms that a user can configure by means of site-specific extraction rules. The literature provides a variety of techniques to infer these rules, which range from regular expressions to context-free grammars, horn clauses, tree templates, or transducers, to mention a few. Unfortunately, none of them is universally applicable, which makes this quite an active research area [4].

Our focus is on information extractors whose extraction rules are transducers. From a theoretical point of view, a transducer is a finite-state machine with an input tape and an output tape, contrarily to regular finite automata, which have a single input tape. In our context, the input tape consists of an input HTML page, which is represented as a sequence of characters, and the output tape is a sequence of slots that hold part of the information in the input tape in a structured format. Information to be extracted from web pages depends on the context. The benefit of using transducers instead of other types of rules is that they can easily deal with repeating, alternating or optional input patterns.

There are a number of techniques in the literature to infer transducers from sample web pages [5][10][8][3]. They all are supervised techniques, i.e., they require a user to annotate several web pages to create a training dataset. The annotation process, c.f. Figure 1, consists of tagging the relevant information fragments to be extracted, e.g., using special tags, an external offsets file, or an external XPaths file. The goal of the learning process is to learn a transducer that transforms pages, that are similar in structure to the pages in the training dataset, into sequences of structured slots that represent part of the information in the input pages.

Unfortunately, there does not seem to be a common transducer model in the literature of information extraction. The techniques available have been developed in isolation, which has resulted in several transducer models in current use. In this paper, we present a new transducer model that unifies existing models. We have devised a general algorithm that allows to infer the skeleton of our transducers from a training dataset, and we have adapted three techniques in the literature to learn state transitions. Obtained results show that our model can be used by several rule learning techniques and it can achieve high effectiveness in the domain of information extraction.

This paper is organised as follows: First, Section II surveys briefly some proposals on information extraction that used transducers. Section III presents our transducers model for information extraction using formal language. Section IV describes how transducers are created and how different rule learning algorithms can be adapted for this purpose. Experimental results are reported in Section V to validate our model. We conclude our work in Section VI.

II. Related Work

Information extractors are used to extract and structure information from unstructured web pages, such as news and blogs, or from semi-structured web pages, such as web pages with one or more result records and detail web pages with information about a certain product. Our work focuses on proposals for information extraction from semi-structured web pages.

The literature has provided several proposals on information extraction that are based on transducers. Furthermore, another proposals can be adapted easily by reusing their learning algorithms to learn transition conditions of a transducer. We survey briefly these proposals:

- Softmealy [5] constructs a transducer in which each state indicates an attribute to extract and the order in which the annotations appear in the input web pages. Separating

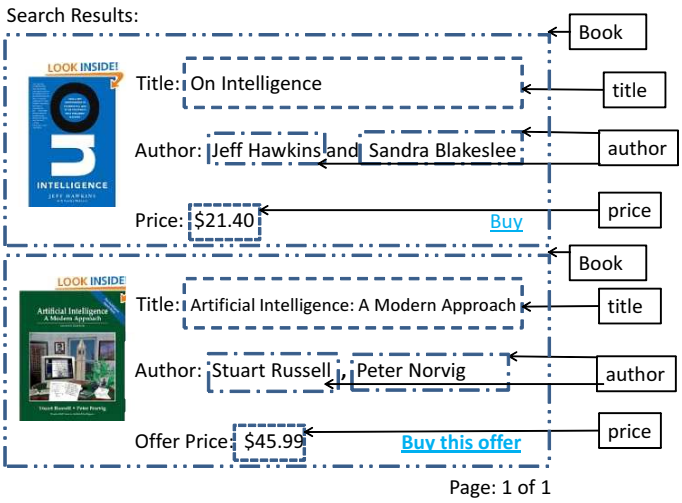


Fig. 1. Annotations performed by user.

text for each transition is obtained from input web pages and then aligned and generalised to learn the transition conditions. It uses a hierarchical tokenisation and an alignment algorithm.

- WIEN [9] identifies six learning classes. These classes work on learning the maximum common prefix and suffix for each type of annotation and then interpret them as regular expressions. Some of these classes extract flat attributes, while others extract nested data. In this work we use LR learning class which detects the largest common prefix and suffix tokens.
- FiVaTech [6] works on learning a template tree that defines the information present on a web page. It uses a clustering technique to label HTML tree nodes and then applies alignment and pattern detection techniques over the sequences of nodes. Then, FiVaTech works on detecting the data scheme used in these pages by detecting tuple types and the order in which information appears.

Another proposal on transducers for information extraction is the one proposed in [3]. This technique described how traditional transducers can be considered as information extractors, but does not propose any rule learning technique. The difference with our model is that ours is tailored specifically to information extraction, and labelling of extracted information, and allows adapting several rule learning techniques for this purpose.

III. Transducer Model

A. Formal Definition

A transducer is a tuple of the form (S, T, i, f, M), where S denotes a finite set of states, T a finite set of transitions, $i \in S$ the initial state, $f \in S$ the final state, and M is a map that associates a user-defined label with every state. Transitions are tuples of the form $(a, L_1 \mid R_1 \rightarrow L_2 \mid R_2, b)$, where $a \in S$ is the source state, $b \in S$ is the target state, and $L_1 \mid R_1 \rightarrow L_2 \mid R_2$ is a condition in which L_1, R_1, L_2, and R_2 are regular expressions. If $a = i$, then $L_1 = \hat{}$, which is a regular expression that denotes the beginning of a string; if $b = f$, then $R_2 = \$$, which is a regular expression that denotes the end of a string.

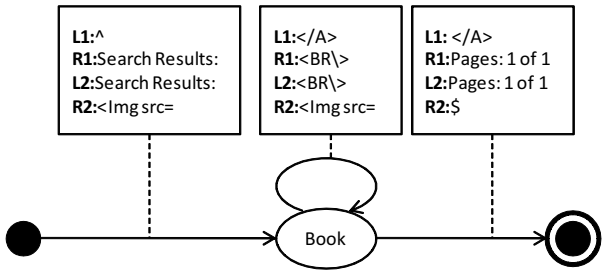


Fig. 2. An example of a transducer to extract Books learnt by a specific rule transition technique from annotations in Figure 1.

Transducers run on configurations of the form (W, R, c, p), where W is the sequence of characters in an input web page, R is a sequence of slots, c denotes the current state, and p denotes the current offset in W. Information extraction takes place at every state, excluding i and f; map M is used to assign a label to every piece of information that is extracted in a state. By slot we mean a piece of text that is tagged with a user-defined label; note that a slot may refer to an attribute, e.g., title, author, or price, or to a record, which is a piece a text to which another transducer must be apply in order to extract its attributes, e.g., Book.

At run time, every transducer starts in a configuration of the form $(W, \langle\rangle, i, 0)$. Given an arbitrary configuration of the form (W, R, c, p) a transition of the form $(c, L_1 \mid R_1 \rightarrow L_2 \mid R_1, c')$ may take place as long as the following conditions hold:

- L_1 matches W at position $pos_{L_1} \geq p$); we denote the length of this match as len_{L_1};
- R_1 matches W at position $pos_{R_1} = pos_{L_1} + len_{L_1}$; we denote the length of this match as len_{R_1};
- L_2 matches W at position $pos_{L_2} \geq pos_{L_1} + len_{L_1}$; we denote the length of this match as len_{L_2}.
- R_2 matches W at position $pos_{R_2} \geq pos_{L_2} + len_{L_2}$; we denote the length of this match as len_{R_2}.

In case the previous transition takes place, the configuration is transformed into (W, R', c', p'), where $R' = R \frown \{M(c) \mapsto W[p \,..\, pos_{L_1} + len_{L_1} - 1]\}$ if $c \neq i$, $R' = R$ if $c = i$, and $p' = pos_{L_2} + len_{L_2} - 1$; $\alpha \frown \beta$ represents the concatenation of sequences α and β, and $\alpha[i \,..\, j]$ the subsequence of α between indices i and j.

B. Disambiguation

Consider the following transducer (S, T, i, f, M) and the configuration (W, R, c, p) where c is the source state for n transitions where $n > 1$, these transitions are as follows:

- $T_1 : (c, L_1 | R_1 \rightarrow L_2 | R_2, c')$
- $T_2 : (c, L'_1 | R'_1 \rightarrow L'_2 | R'_2, c'')$

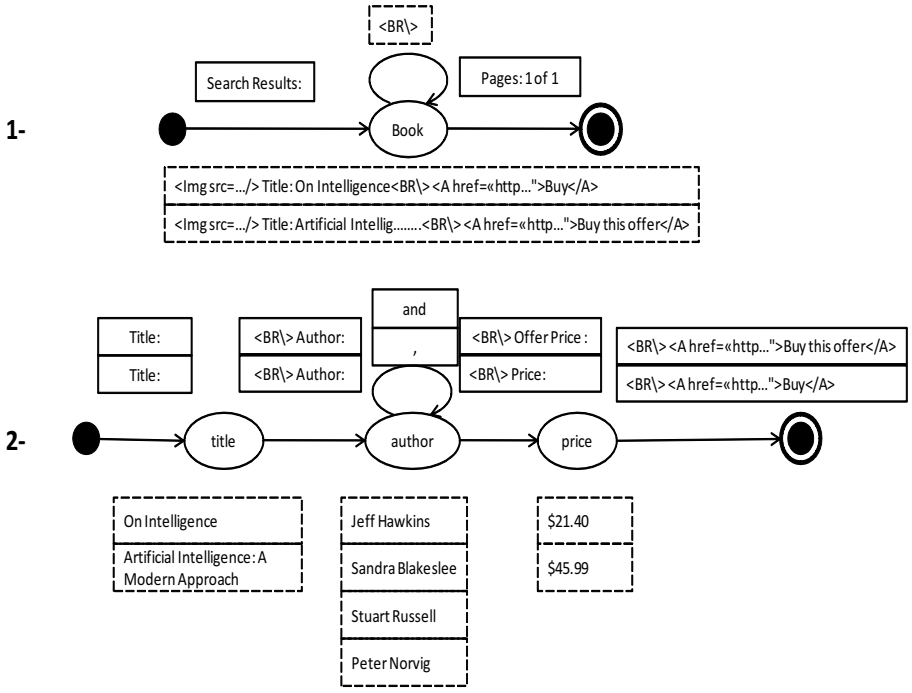


Fig. 3. Skeleton learnt from the annotated examples in Figure 1

.

- $T_n : (c, L_1^n | R_1^n \rightarrow L_2^n | R_2^n, c^n)$

When more than one transition conditions hold, an ambiguity should be resolved in order to select which transition should be performed. A set of heuristics were studied to select the next transition:

- Select the first transition that matches without checking the other ones.
- Assigning to each transition a weight and sorting them in a descendent way. This weight is obtained by calculating the number of cases in the training dataset where this transition took place.
- Sort the transitions according to the matching index of L_1—R_1 condition. If the matching index is equal in all cases, they are sorted by keeping the transitions to the final states at the end of these possible transitions.

While the first heuristic is arbitrary which causes results to vary depending on which transition is chosen at each moment, the second one gives priority to usual transitions while the third one considers the earliest matchings more important. According to our experimental results, the last heuristic performs better than the first two ones.

C. Transducer Example

Figure 2 shows a transducer learnt from the annotated examples in Figure 1. Its transition conditions learnt using a rule learning technique from literature. When the conditions of the transition from the initial state to the state labelled with $Book$ tag hold, then the transducer consumes characters until reaching the first Book and the current state changes to this state. It now tests which of the two transitions, whose source is the current state, hold. In case the conditions of the transition from Book to Book hold, a Book is extracted and the pointer now points at the start of the next book. In case the conditions of the transition whose target is the final state hold, a book is extracted and the transducer consumes until the end of the input text. In case there is an ambiguity and the two transitions whose source is the state "Book", one of the previous heuristics can be applied to select and perform a transition.

IV. Learning Transducers

A. Learning a Skeleton

A skeleton contains a transducer that does not have transition conditions and all the information from the training dataset that originated the states and transitions in the transducer. It is the learning technique responsibility to learn these conditions. The skeleton is learnt starting from a learning dataset and it describes the structure, type and order of relevant information on the web pages from the learning dataset.

A skeleton is a tuple of the form $\delta(T, \mu, \iota)$, where T is a transducer, μ is map in which for each transition t in the the transducer T, it saves a list of all the separating texts that originated this transition. ι is a map that saves for each state s in the transducer T, a list of all the annotations that originated this state.

Figure 3 shows two skeletons learnt starting from the annotated examples in Figure 1. The first contains an uncomplete transducer created to extract book records, but it needs a rule learning algorithm to learn its transition condition. ι in the

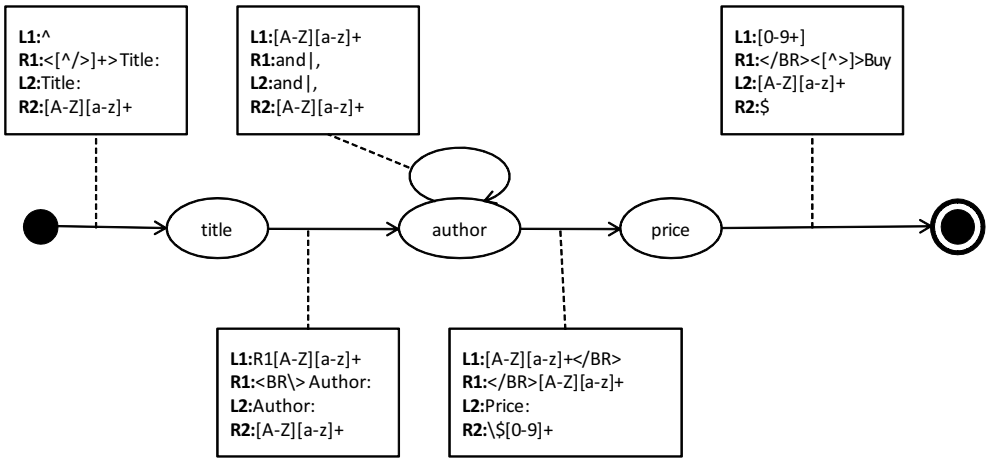


Fig. 4. Transducer learnt using SM learning technique.

first skeleton contains a map with a unique entry for the state with the label "Book" that contains a list of two elements which are the text fragments of the two annotated books. μ has 3 entries, one for each transition. Each entry in μ contains one text fragment since this transition appeared once in the annotated examples. The second skeleton in Figure 3 is learnt to extract Book attributes from text fragments extracted by the "Book" state in the first skeleton.

B. Learning Transition Conditions

Once a skeleton is learnt from a learning dataset, a learning algorithm is now necessary to learn transition conditions. Different learning techniques can be adapted for this purpose counting on the separators and annotations saved in a learnt skeleton. Consider the skeleton learnt in Figure 3, it is possible to learn transition conditions using techniques to infer regular expressions starting from a collection of strings. We adapted three techniques in the literature to prove the flexibility of our transducer model. These techniques are:

- NLR: It was inspired by WIEN [9] proposal, and concretely, from the LR class inside this WIEN proposal. The LR class searches for the longest common prefix and suffix for each annotation and converts it to a regular expression. In our case, L_1 condition was learnt by searching for the common tokens at the end of the individuals in the origin state. R_1 condition was learnt by searching for the common tokens at the beginning of all the separating texts for the transition. Common tokens at the end of these separators are used to learn L_2 and the common tokens at the beginning of the individual texts in the target state are used to learn R_2. Figure 2 shows a transducer to extract books learnt applying NLR technique.
- SM: This technique is inspired by SoftMealy [5]. It uses an alignment and generalising algorithms. It only considers the tokens at the beginning and at the end of the separators and of the individuals in each state that are limited by tokens of type WORD. Tokens of type WORD are tokens that are digits and characters, excluding spaces, punctuation and HTML tags. A multiple string alignment and a generalisation algorithms are applied on each set of the considered tokens to learn transition conditions.Figure 4 shows a transducer to extract attributes inside Books learnt applying our SM technique.
- FT: It is inspired by a multiple string alignment technique used in FiVaTech [6]. Separators in each transition are aligned using the multiple sequence alignment algorithm used in FiVaTech to create regular expressions and learn transition conditions.

V. Experimental Results

To perform our experiments, we first created a collection of datasets from current web sites and followed the guidelines from [7] to perform our empirical comparison.

Table I reports on the results of applying these techniques in practice on several datasets compared side by side. Each dataset contains 30 annotated web pages and a 10-folding cross validation was performed to calculate precision (P) and recall (R). The time column reports the time necessary to learn a transducer for each dataset. The experiments were run on a Windows Server 2008 (64-bits) machine that was equipped with a four core Intel Xeon 3.00 GHz CPU, 16 GB RAM, and JRE 1.6.0.

Sites to create the datasets were chosen arbitrarily and the web pages inside each dataset were chosen to cover all possible data variability, such as Books with one or more authors and Books with and without price. Note that these techniques can obtain better precision and recall by adding more web pages to these learning datasets, but this is not our study case since we are just checking if implemented techniques can achieve good results and to validate our transducers model. Better results may be obtained if each one of the used web sites are studied in-depth to choose the web pages to include in the web site's dataset.

TABLE I
COMPARING PRECISION AND RECALL OF NLR, SM AND FT TECHNIQUES USING TRANSDUCERS.

Dataset	NLR			SM			FT		
	P	R	Time	P	R	Time	P	R	Time
netlib	0.433	0.393	0.312m	0.853	0.520	0.544m	0.862	0.176	1.210m
albanianfilmdatabase.com	0.874	0.245	0.700m	0.874	0.304	0.120m	0.962	0.371	0.610m
disneymovieslist.com	0.731	0.731	0.900m	0.989	0.460	0.150m	1.000	0.000	0.930m
imdb.org	0.753	0.855	2.300m	0.985	0.845	0.630m	1.000	0.124	2.310m
citwf.com	0.915	0.915	4.800m	0.981	0.878	0.130m	0.992	0.892	4.810m
betterworldbooks.com	0.993	0.915	0.530m	0.877	0.844	0.411m	0.920	0.514	0.747m
manybooks.net	0.974	0.824	0.480m	0.746	0.067	0.550m	0.770	0.536	0.311m

VI. CONCLUSIONS

We have proposed a new powerful model of transducers for web information extraction. To validate our model, we have adapted several rule learning algorithms which shows the feasibility and flexibility of our model. The model has high expressiveness since it supports disjunction, multi-order attributes, optional and multi-valued attributes. The experimental results demonstrate the high effectiveness that can be achieved on current web sites using our model which is almost 100% in some cases.

ACKNOWLEDGMENT

This paper was supported by the European Commission (FEDER), the Spanish and the Andalusian R&D&I programmes (grants TIN2007-64119, P07-TIC-2602, P08-TIC-4100, TIN2008-04718-E, TIN2010-21744, TIN2010-09809-E, TIN2010-10811-E, and TIN2010-09988-E).

REFERENCES

[1] Manuel Álvarez, Alberto Pan, Juan Raposo, Fernando Bellas, and Fidel Cacheda. Extracting lists of data records from semi-structured web pages. *Data Knowl. Eng.*, 64(2):491–509, 2008.

[2] Chia-Hui Chang, Mohammed Kayed, Moheb R. Girgis, and Khaled F. Shaalan. A survey of web information extraction systems. *IEEE Trans. Knowl. Data Eng.*, 18(10):1411–1428, 2006.

[3] Boris Chidlovskii. Wrapping web information providers by transducer induction. In *European Conference on Machine Learning*, pages 61–72, 2001.

[4] AnHai Doan, Jeffrey F. Naughton, Raghu Ramakrishnan, Akanksha Baid, Xiaoyong Chai, Fei Chen 0002, Pedro DeRose, Byron J. Gao, Warren Shen, Ting Chen, Ba-Quy Vuong, Eric Chu, Chaitanya Gokhale, and Jiansheng Huang. Information extraction challenges in managing unstructured data. *SIGMOD Record*, 37(4):14–20, 2008.

[5] Chun-Nan Hsu and Ming-Tzung Dung. Generating finite-state transducers for semi-structured data extraction from the web. *Inf. Syst.*, 23(8):521–538, 1998.

[6] Mohammed Kayed and Chia-Hui Chang. FiVaTech: Page-level web data extraction from template pages. *IEEE Trans. Knowl. Data Eng.*, 2010.

[7] Barbara Kitchenham, Shari Lawrence Pfleeger, Lesley Pickard, Peter Jones, David C. Hoaglin, Jarrett Rosenberg, and Khaled El Emam. Preliminary guidelines for empirical research in software engineering. *IEEE Trans. Software Eng.*, 28(8):721–734, 2002.

[8] Raymond Kosala, Hendrik Blockeel, Maurice Bruynooghe, and Jan Van den Bussche. Information extraction from structured documents using k-testable tree automaton inference. *Data Knowl. Eng.*, 58(2):129–158, 2006.

[9] Nicholas Kushmerick. Wrapper induction: Efficiency and expressiveness. *Artif. Intell.*, 118(1-2):15–68, 2000.

[10] Ion Muslea, Steven Minton, and Craig A. Knoblock. Hierarchical wrapper induction for semistructured information sources. *Autonomous Agents and Multi-Agent Systems*, 4(1/2):93–114, 2001.

[11] Nikolaos Papadakis, Dimitrios Skoutas, Konstantinos Raftopoulos, and Theodora A. Varvarigou. Stavies: A system for information extraction from unknown web data sources through automatic web wrapper generation using clustering techniques. *IEEE Trans. Knowl. Data Eng.*, 17(12):1638–1652, 2005.

On improving FOIL Algorithm

Patricia Jiménez, J.L. Arjona, J.L. Álvarez
The Distributed Group - Onuba
Escuela Politécnica Superior.
Crta. Huelva - La Rábida. Palos de la Frontera 21071
Email: {patricia.jimenez, arjona, alvarez}@dti.uhu.es

***Abstract*—FOIL is an Inductive Logic Programming Algorithm to discover first order rules to explain the patterns involved in a domain of knowledge. Domains as Information Retrieval or Information Extraction are handicaps for FOIL due to the huge amount of information it needs manage to devise the rules. Current solutions to problems in these domains are restricted to devising ad hoc domain dependent inductive algorithms that use a less-expressive formalism to code rules.**

We work on optimising FOIL learning process to deal with such complex domain problems while retaining expressiveness. Our hypothesis is that changing the information gain scoring function, used by FOIL to decide how rules are learnt, can reduce the number of steps the algorithm performs. We have analysed 15 scoring functions, normalised them into a common notation and checked a test in which they are computed. The learning process will be evaluated according to its efficiency, and the quality of the rules according to their precision, recall, complexity and specificity. The results reinforce our hypothesis, demonstrating that replacing the information gain can optimise both the FOIL algorithm execution and the learnt rules.

***Index Terms*—FOIL, ILP, scoring functions**

I. Introduction

Machine learning systems aim to automatically learn to recognize complex patterns based on data from some background knowledge and to make intelligent decisions on new data. Many of these systems has their focus on Inductive Logic Programming (ILP), a subfield of machine learning which investigates the construction of first-order logic rules. This kind of systems include FOIL [20], GOLEM [13], PROGOL [12] with Shapiro's program MIS as one of their early predecessors [11].

However, a major problem of ILP systems arises when the set of training data is too large. Obviously, it happens in domains as information retrieval or information extraction; the process of learning the hypothesis that best fits the available knowledge becomes inefficient, or the set of learnt rules has low recall. An alternative choice is propositional logic systems since they result more practical for efficiency reasons, but they produce rules quite less expressive and consequently, they are restricted to be applied to simpler domains problems.

We wish to use FOIL algorithm to deal with such complex domain problems and we try to improve it in order make the learning process more efficient. To carry out this task, we have studied 15 different scoring functions coming from statistics, machine learning, and data mining literature and we propose to use the best one instead of the information gain, which is employed by the original FOIL to select the best candidate rules. We have proved that some of these scoring functions perform better since they find out best rules or find them out faster.

We bet on FOIL algorithm because of the expressiveness of first-order logic rules it is able to devise. First-order rules allow the system to learn relational and recursive concepts that cannot be represented in the attribute-value format assumed by most machine learning algorithms. Furthermore, there have been many authors who have tried to improve FOIL developing successors systems like FOCL [25], AUDREYII [18], mFOIL [17], HYDRA [16], FOSSIL [15], FFOIL [14], FZ-FOIL [27] and FOIDL [10]. Some of these systems proposed to use likelihood ratio, correlation criterion, estimated accuracy and interest measures as alternatives to the information gain. In many cases, their results were better than FOIL ones but restricted to domain dependent tasks. So that the problem has not been solved yet. However, these systems and their results suggest that FOIL can be optimised in many ways.

The paper is organised as follows: first, we introduce an overview of FOIL algorithm and we propose to use new scoring functions in order to solve the mentioned problems. Next section a common notation and a set of scoring functions are explained. Then, we perform a test we show the results obtained. Conclusion section discuss these results and gives some tips for future research.

II. FOIL

In first order learning, training data comprises a target predicate, which is defined by a set of ground literals labelled as positive, if they satisfy the target predicate, and as negative, otherwise. Furthermore, a set of support predicates is defined either extensionally, similarly to what was previously made with the target predicate or intensionally, by means of a set of rules. The goal is to learn a set of logic rules that explain the target predicate in terms of itself and the support predicates.

FOIL is an algorithm of machine learning that induces first order rules. It is based on sequential covering algorithm and uses separate-and-conquer method, attempting to learn one rule at a time to incrementally grow the final set of rules. In order to learn each rule, it follows a top-down approach, starting with the most general rule header, and guided by a greedy search, is adding new unground literals to the rule, until it does not satisfy any negative ground literal belonging to the target predicate. The set of rules is ready when all positive ground literals belonging to the target predicate are satisfied.

Each learnt rule is of the form $H \leftarrow B$ where H is the head and B is the body of the rule. H is an unground literal of the form $R(X_0, X_1, \ldots, X_n)$ where R is the target predicate and $X_0, X_1, \ldots, X_n$ are the variables. Similarly, B is a set of unground literals, for instance $P_1(X_0, X_2), P_2(X_3, X_1), \ldots$, where P_i represents any predicate defined in the knowledge base and $X_0, X_1, \ldots, X_n$ are the variables of the predicate P_i.

To add a new literal to the current rule, a list of unground literals is generated. Each one is added to the current rule giving rise to a new candidate rule. The candidate rules are weighted based on information gain scoring function. It measures how benefits replacing the current rule with a specific candidate rule. To compute this score, the information gain relies on the number of positive and negative ground literals that are satisfied before and after this replacement. The candidate rule with higher score is selected to keep growing.

Let tp be the number of positive ground literals and fp the number of negative ground literals that are satisfied by the current rule. The information conveyed by the knowledge that a ground literal satisfied by the current rule is positive is given by

$$I(H \leftarrow B) = -log\frac{tp}{tp + fp} \tag{1}$$

Similarly, for each new candidate rule $I_k(H \leftarrow B')$ built from adding a new literal generated L_k to the current rule. Being t the number of positive ground literals satisfied by both the current and a new candidate rule, the information gain has a straightforward interpretation in terms of information theory and is given by the formula:

$$I(H \leftarrow B') = t \times (I_k(H \leftarrow B') - I(H \leftarrow B)) \tag{2}$$

In FZFOIL [27] some deficiencies in the information gain have been identified. Presumably, it may be due to the information gain only take into account the number of positive and negative ground literals a new candidate rule satisfies, forgetting other parameters as the number of positive and negative ground literals this candidate rule discards.

For the purpose of improving the learning process, we analyse other scoring functions from the literature trying to solve the information gain problem stated. They weigh the candidate rules up according to the existing correlation between it and the current rule. Therefore, the gain of these scoring functions will measure the amount of correlation gained if the current rule is replaced with a new specific candidate rule.

III. Comparison Framework

The proposed scoring functions will be defined in terms of the well-known contingency table. For evaluating any first order candidate rule $H \leftarrow B'$, we rely on a contingency table as the one below in I.

Actual class are those positive and negative ground literals satisfied by the head of a candidate rule. Predicted class are those ground literals satisfied by the body of the candidate rule being analysed. Thus, tp denotes the number of positive ground literals that are satisfied by the head and the body of the candidate rule and fp denotes the number of negative ground literals satisfied by the head and the body of the candidate rule. Similarly, fn denotes the number of positive ground literals satisfied by the head but not by the body of the candidate rule and tn denotes the number of negative ground literals satisfied by the head but not by its body. N is the total number of positive and negative ground literals.

		Predicted Class		
		B	$\neg B$	
Actual Class	H	tp true positives	fn false negatives	
	¬ H	fp false positives	tn true negatives	
				N

TABLE I
Contingency Table

	Scoring Function	Formula
1	Coverage	$\frac{tp+fp}{N}$
2	Laplace Accuracy	$\frac{tp+1}{tp+fp+2}$
3	Leverage	$\frac{tp \cdot tn - fn \cdot fp}{N^2}$
4	ϕ-coefficient	$\frac{tp \cdot tn - fn \cdot fp}{\sqrt{(tp+fn) \cdot (tp+fp) \cdot (fp+tn) \cdot (fn+tn)}}$
5	Support	$\frac{tp}{N}$
6	Confidence	$\frac{tp}{tp+fp}$
7	Satisfaction	$\frac{tp \cdot tn - fp \cdot fn}{(tp+fp) \cdot (tn+fp)}$
8	Confirmation	$\frac{(tp \cdot tn - fp \cdot fn)^2}{N^2 \cdot (tp+fp) \cdot (tn+fp)}$
9	F-measure	$2 \cdot \frac{tp}{2 \cdot tp + fn + fp}$
10	kappa (κ)	$\frac{2 \cdot (tp \cdot tn - fp \cdot fn)}{N^2 - (tp+fn) \cdot (tp+fp) - (fp+tn) \cdot (tn+fn)}$
11	Odds-ratio	$\frac{tp \cdot tn}{fp \cdot fn}$
12	Yule's Q	$\frac{tp \cdot tn - fp \cdot fn}{tp \cdot tn + fp \cdot fn}$
13	Lift (Interest)	$\frac{N \cdot tp}{(tp+fp) \cdot (tp+fn)}$
14	Collective Strength	$\frac{tp+tn}{(tp+fp) \cdot (tp+fn) + (fn+tn) \cdot (fp+tn)}$ $\times \frac{N^2 - (tp+fp) \cdot (tp+fn) - (fn+tn) \cdot (fp+tn)}{N - tp - tn}$
15	Jaccard(ζ)	$\frac{tp}{tp+fp+fn}$

TABLE II
List of Scoring functions

We have implemented and evaluated a subset of scoring functions proposed in [24] as objective measures and in [19] as measures for predictive and descriptive induction. Furthermore, we have selected other scoring functions for being quite traditional. The set of scoring functions adapted to our notation are showed in table II.

A summary description for each scoring function:

- Coverage is a measure of generality of a rule. If a rule characterizes more information in the data set, it tends to be more interesting.
- Laplace Accuracy [23] is an approximate measure to estimate the expected accuracy directly. General rules tend to be favored.
- Leverage [22] is one of the most frequently measure used in the evaluation of rules. It is also known as Leverage. It trades off generality and relative accuracy.
- $\phi - coefficient$ [21] is a statistical measure analogous to Pearson's product-moment correlation coefficient. It measures the degree of association between two binary variables (e.g., two rules). It is closely related to the χ^2 statistic since $\phi^2 = \frac{\chi^2}{N}$.
- Support [9] is a measure known from association rule learning, also called frequency. It is used for specifying if a rule is observed frequent enough in a data set.
- Confidence is also known as confidence [9]. It is related to the reliability. A rule is reliable if its predictions are highly accurate.
- Satisfaction [19] is similar to confidence e.g., $Sat(H \leftarrow B) = 1$ if $Confidence(H \leftarrow B) = 1$, but, unlike Confidence, it takes the entire contingency table into account and is thus more suited towards knowledge discovery.
- Confirmation [8] is defined in terms of a modified χ^2 statistic. It trades off satisfaction and Leverage measures.
- F-measure is other statistic measure of a test accuracy. It considers both the precision and the recall of the test to compute the score. We compute F1-score which is the harmonic mean of precision and recall.
- kappa (κ) [7] captures the degree of agreement between a pair of variables (e.g., the head and the body of a candidate rule). If both variable are highly agree with each other, then the values for κ will result higher.
- Odds-ratio [5] represents the strength of association or non-independence between two binary data values. Unlike other measures of association for paired binary data, the comparison between the two variables is symmetrical.
- Yule's Q coefficient [6] is a normalized variant of the odds ratio.
- Lift (Interest) [3] is used quite extensively in data mining for measuring deviation from statistical independence. It gives an indication of rule significance or interest.
- Collective Strength [4] is other measure of correlation variant of Lift measure. It compares between actual and expected values.
- Jaccard(ζ) [2] is a statistic used for comparing the similarity and diversity of sample sets. It is used extensively in information retrieval to measure the similarity between documents. We measure the similarity between two rules.

We have defined some measures to help us decide which scoring function is more promising. The set of measures taking into account are:

1) ***Efficiency***. This is defined as the amount of useful work in relation to time and resources used. The resources are memory and space required.
2) ***Precision***. It measures the number of ground literals satisfied by the set of rules correctly against all ground literals satisfied, although so far, we only search for 100% accuracy rules, i.e., we do not allow rules that satisfy any negative ground literal.
3) ***Recall***. It determines if a set of rules is complete, i.e., if it satisfies all positive ground literals belonging to the target predicate.
4) ***Complexity*** of the induced set of rules. It is computed in terms of bits from Minimum Description Length Principle [26].
5) ***Specificity/Generality*** of the induced set of rules. It is general if they are only a few single rules that satisfy most of positive ground literals belonging to the target predicate. They will be too specificity when they are too large and only satisfy a few number of positive ground literals. We prefer general rules rather specific rules.

However, there are measures that can not be estimated objectively because they depend on other measures which we call secondary measures. For instance, secondary measures that may affect efficiency directly may be the number of backtracking performed or the number of candidate rules that were evaluated, which is one of the most expensive step in the algorithm. The number of different predicates used in the induced set of rules could also affect the efficiency of the learning process because it gives an idea about how well the knowledge base was built and therefore, how useful the support predicates defined are.

Secondary measures that may affect to generality/specificity of a rule are the number of the variables used in the set of rules and the deep of the learnt rules measured in terms of the number of unground literals in each rule. If these numbers are small it will mean that the rules are quite general, which is a desirable property.

Intuitively, the total number of induced rules will affect both, efficiency and generality/specificity. Fewer number of rules will make the process more efficient and the final set of rules more general. Note the latter is true as long as the set of rules has a good recall.

IV. On Going Work

The example tested was first showed in [27] and it tries to explain when somebody is sick. To carry out this task we rely on a set of seventeen individuals among which eight are sick, and the rest are not. The target predicate will be sick(X_0), which means the individual X_0 is sick. The support predicates defined to induce a set of rules that explain the target predicate sick(X_0) are:

- bearded(X_i), which means the individual X_i is bearded.
- smoker(X_i), which means the individual X_i is a smoker.
- father(X_i, X_j), which means X_i is X_j's father.
- boss(X_i, X_j), which means X_i is X_j's boss.

In the knowledge base, the positive ground literals belonging to the target predicate are all individuals who are sick. The rest are the negative ground literals and they can be defined explicitly or to be induced by Closed World Assumption. Similarly, we have to define the positive ground literals that satisfy each support predicate but there is no need to define the negative ones explicitly.

Scoring Function	P	R	C	T
Information Gain	1	1	23.75	6257
Coverage	1	0.5	12.17	138275
Laplace Accuracy	1	1	30.66	22317
Leverage	1	1	22.34	5400
Φ-coefficient	1	1	37.09	34200
Support	1	1	26.77	43759
Confidence	1	1	23.08	19017
Satisfaction	1	1	20.17	3512
Confirmation	1	1	20.17	3705
F-Measure	0	0	0	58147
Kappa	1	1	22.34	4983
Odds ratio	1	1	25.92	10470
Yule's Q	1	1	56.85	25701
Lift	1	1	23.08	17249
Collective Strength	1	1	22.34	4745
Jaccard	0	0	0	57480

TABLE III
RESULTS OBTAINED

Our knowledge base would be a Prolog program and the set of learnt rules for this example is showed in IV[1]. The results obtained for the evaluation measures explained previously, are presented in table IV[2]

V. CONCLUSIONS

As well as in other analysis of measures or scoring functions, we can not conclude saying there is a scoring function consistently better than the rest in all applications domains, although we have found some scoring functions that perform better than information gain in our running example.

All rules induced are 100% accurate because we do not allow rules that satisfy any negative ground literal. Therefore the goal is to get the most reduced set of learnt rules with the largest recall in the shortest time possible. The final set of rules will depend largely on the scoring function used. If it is not good enough FOIL might not learn a complete set of rules (i.e, the set do not have a recall of 100%). We wish a balance among precision, recall, efficiency, complexity and specificity/generality to decide which is the most promising set of rules obtained.

Taking of these factors into account, we consider that Collective Strength and Leverage scoring function performed better maintaining the full recall, because they took shorter time to get the rules. Furthermore the set of rules were more general and less complex. Satisfaction, Confirmation and kappa scoring functions are even better than the previous one. They took less time to find out a set of rules and, although Satisfaction and Confirmation scoring functions had one rule more, both had a complexity still lower.

Support scoring function is quite similar to the information gain. It spent more time evaluating many candidate rules and the rules are more complex but more general. It is difficult

[1]Note that FOIL relies on predefined predicates which are of the form $X_i = X_j$ or $X_i = c_j$, where X_i and X_j are variables and c_j is a constant (e.g., c_j can be any specific individual).

[2]where: **P**: Precision, **R**: Recall, **C**: complexity (bits), **T**: elapsed time (milliseconds)

Rules
Information Gain
sick(X_0)← smoker(X_0).
sick(X_0)← boss(X_1,X_0), sick(X_1), ¬father(X_0, X_2).
sick(X_0)← boss(X_1,X_0), sick(X_1), father(X_2, X_0), sick(X_2).
Coverage
sick(X_0)← smoker(X_0).
sick(X_0)← boss(X_0,X_1), father(X_0, X_2).
Laplace Accuracy
sick(X_0)← father(X_1,X_0), bearded(X_1), smoker(X_0).
sick(X_0)← boss(X_1,X_0), sick(X_1), father(X_2,X_0), sick(X_2).
sick(X_0)← ¬father(X_0,X_1), boss(X_1,X_2), boss(X_2,X_3), boss(X_4,X_1), father(X_5,X_2), $X_0 \neq X_3$.
Leverage
sick(X_0)← father(X_1,X_0), bearded(X_1), boss(X_0,X_2).
sick(X_0)← ¬father(X_0,X_1), boss(X_1,X_2), ¬boss(X_2,X_0).
ϕ-coefficient
sick(X_0)← ¬boss(X_0,X_1), boss(X_1,X_2), ¬boss(X_2,X_0).
sick(X_0)← father(X_1,X_0), boss(X_0,X_2), ¬father(X_0,X_3).
sick(X_0)← father(X_1,X_0), bearded(X_1), boss(X_0,X_2).
Support
sick(X_0)← boss(X_0,X_1), smoker(X_0).
sick(X_0)← father(X_1,X_0), boss(X_1,X_2), $X_1 \neq X_2$, boss(X_1, X_3), sick(X_1).
Confidence
sick(X_0)← smoker(X_0).
sick(X_0)← boss(X_1,X_0), sick(X_1), father(X_2,X_0), sick(X_2).
sick(X_0)← ¬father(X_0,X_1), boss(X_1,X_2), boss(X_2,X_3), boss(X_4,X_1), father(X_5,X_2), $X_0 \neq X_3$.
Satisfaction
sick(X_0)← smoker(X_0).
sick(X_0)← ¬father(X_0,X_1), ¬bearded(X_0).
sick(X_0)← boss(X_1,X_0), sick(X_1), bearded(X_1).
Confirmation
sick(X_0)← smoker(X_0).
sick(X_0)← ¬father(X_0,X_1), ¬bearded(X_0).
sick(X_0)← boss(X_1, X_0), sick(X_1), bearded(X_1).
F-measure
kappa (κ)
sick(X_0)← father(X_1,X_0), bearded(X_1), boss(X_0,X_2).
sick(X_0)← ¬father(X_0,X_1), boss(X_1,X_2), ¬boss(X_2,X_0).
Odds-ratio
sick(X_0)← smoker(X_0).
sick(X_0)← boss(X_1,X_0), boss(X_0,X_2), sick(X_2), sick(X_1).
sick(X_0)← ¬boss(X_0,X_1), boss(X_1,X_2), ¬bearded(X_0).
Yule's Q
sick(X_0)← ¬bearded(X_0), boss(X_1,X_0), sick(X_1).
sick(X_0)← boss(X_1,X_0), ¬father(X_1,X_2).
sick(X_0)← smoker(X_0).
sick(X_0)← father(X_1,X_0), ¬boss(X_2,X_0).
sick(X_0)← boss(X_0,X_1), boss(X_2,X_0), smoker(X_1).
sick(X_0)← boss(X_0,X_1), boss(X_2,X_0), ¬father(X_0,X_3).
Lift (Interest)
sick(X_0)← smoker(X_0).
sick(X_0)← boss(X_1,X_0), sick(X_1), father(X_2,X_0), sick(X_2).
sick(X_0)← ¬father(X_0,X_1), boss(X_1,X_2), boss(X_2,X_3), boss(X_4,X_1), father(X_5,X_2), $X_0 \neq X_3$.
Collective Strength
sick(X_0)← father(X_1,X_0), bearded(X_1), boss(X_0,X_2).
sick(X_0)← ¬father(X_0,X_1), boss(X_1,X_2), ¬boss(X_2,X_0).
Jaccard(ζ)

TABLE IV
SET OF LEARNT RULES

to decide which is better. As efficient is a relevant factor, we would opt for information gain. However, Laplace accuracy, ϕ-coefficient, Confidence and Lift behaved worse than information gain, wasting time searching for more specific rules. Note Lift and Laplace accuracy obtained similar results.

Finally, Coverage, FMeasure and Jaccard scoring functions did not find out a set of rules that satisfy all positive ground literals defined in the knowledge base. The last two were unable to find out a single rule and they evaluated a huge amount of candidate rules caused among other factors, by the backtracking performed. Note FMeasure and Jaccard scoring function has very similar formulae so they behaved in an identical way.

We can conclude saying that scoring functions like Leverage, Confirmation, Satisfaction, Kappa and Collective Strength are more promising than the information gain. Anyway, establishing a ranking among the proposed scoring functions is a hard task because we have not identified which are the most relevant evaluation measures yet and there could be more interesting measures to check. The relevance of each one will depend on the domain being studied although we opt for those measures that affect the efficiency directly as more relevant.

The application of this kind of systems is usually better than with any other known approach, so it needs to find more training sets to be tested, to define more additional measures if that would be necessary and to perform an exhaustive evaluation to get a reliable ranking. All this in order to apply FOIL satisfactorily to domains with a huge amount of information.

ACKNOWLEDGMENT

This research is supported by the European Commission (FEDER), the Spanish and the Andalusian R&D&I programmes (grants TIN2007-64119, P07-TIC-2602, P08-TIC-4100, and TIN2008-04718-E).

REFERENCES

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.
[2] C.J. Van Rijsbergen, *Information Retrieval*, Butterworth, 1979.
[3] S. Brin, R. Motwani, C. Silverstein, *Beyond Market Baskets: Generalizing Association Rules to Correlations*, SIGMOD Conference, 1997.
[4] C.C. Aggarwal, P.S. Yu, *A New Framework For Itemset Generation*, Symposium on Principles of Database Systems, 1998.
[5] F. Mosteller, *Association and Estimation in Contingency Tables*, Journal of the American Statistical Association, 1968.
[6] G.U. Yule, *On the Association of Attributes in Statistics*, Philosophical Transactions of the R.Society, Ser. A, 1900.
[7] J. Cohen, *A Coefficient of Agreement for Nominal Scales*, Educational and Psychological Measurement, 1960.
[8] P. Flach, N. Lachiche, *A First-Order Approach to Unsupervised Learning*, 1999.
[9] R. Agrawal, T. Imielinski, A.N. Swami, *Mining Association Rules between Sets of Items in Large Databases*, SIGMOD Conference, 1993.
[10] R.J. Mooney, M.E. Califf, *Induction of First-Order Decision Lists: Results on Learning the Past Tense of English Verbs*, JAIR, 1995.
[11] E.Y. Shapiro, *Algorithmic Program Debudding*, Cambridge, MA: MIT Press, 1983.
[12] A. Srinivasan,S.H. Muggleton, R.D. King, M.J.E Sternberg, *Mutagenesis: ILP experiments in a non-determinate biological domain*, Proceedings of the 4th International Workshop on Inductive Logic Programming, 1994.
[13] S. Muggleton, C. Feng, *Efficient Induction of Logic Programs*, Algorithmic Learning Theory, 1990.
[14] J.R. Quinlan, *Learning First-Order Definitions of Functions*, JAIR, 1996.
[15] J. Fürnkranz, *FOSSIL: A Robust Relational Learner*, European Conference on Machine Learning, 1994.
[16] K.M. Ali, M.J. Pazzani, *HYDRA: A Noise-tolerant Relational Concept Learning Algorithm*, IJCAI, 1993.
[17] N. Lavrac, S. Dzeroski, *Inductive Logic Programming: Techniques and Applications*, Ellis Horwood, 1994.
[18] J. Wogulis, M.J. Pazzani, *A Methodology for Evaluating Theory Revision Systems: Results with Audrey II*, IJCAI, 1993.
[19] N. Lavrac, P.A. Flach, B. Zupan, *Rule Evaluation Measures: A Unifying View*, International Workshop on Inductive Logic Programming. 174–185 (1999)
[20] J.R. Quinlan, *FOIL: A Midterm Report*, European Conference on Machine Learning, 1993.
[21] A. Agresti, *Categorical Data Analysis*, Wiley-Interscience, 1990.
[22] G. Piatetsky-Shapiro, *Discovery, Analysis, and Presentation of Strong Rules*, 1991.
[23] P. Clark, R. Boswell, *Rule Induction with CN2: Some Recent Improvements*, EWSL, 1991.
[24] P. Tan, V. Kumar, J. Srivastava, *Selecting the right objective measure for association analysis*, Inf. Syst., 2004.
[25] M.J. Pazzani, C. Brunk, G. Silverstein, *A Knowledge-intensive Approach to Learning Relational Concepts*, ICML, 1991.
[26] J. Rissanen, *Modeling by Shortest Data Description*, Automatica, 1978.
[27] A.J GÓMEZ, G. FERNANDEZ, *Inducción de definiciones lógicas a partir de relaciones: mejoras en los heurísticos del sistema FOIL*, PRODE, 1992.

A REVIEW ON SENSORY FEEDBACK FOR SEMG BASED PROSTHETIC HANDS

Madhavi Anugolu, *Member , IEEE*, Chandrasekhar Potluri, *Member, IEEE*, Adnan Ilyas, Parmod Kumar, Steve Chiu, *Member, IEEE*, Nancy Devine, Alex Urfer, and Marco P. Schoen, *Senior Member IEEE*

Abstract–*This paper addresses issues concerning the integration of artificial limbs with amputees. From the reported literature, it is evident that the afferent feedback to prosthetic users can help with their ability to control the device. In addition, incorporation of such a feedback system increases the ownership sensation of amputees with their artificial limbs. This paper includes an overview of research findings on the development of non-invasive sensory feedback (afferent) systems for hand amputees. The reviewed stimulators for the afferent system are based on vibration and electrical systems, and a combination of the two methods.*

Keywords: *sEMG, hand prosthetic, sensory feedback*

I. INTRODUCTION

According to the information provided by the National Limb Loss Information Center, [1], in the year of 2005, 1.6 million persons were living in the United States with the loss of a limb. It is estimated that by the year 2050 this number will more than double to 3.6 million.

Madhavi Anugolu is with Measurement and Control Engineering Research Center (MCERC), School of Engineering, Idaho State University, Pocatello, Idaho 83209, USA (email: anugmadh@isu.edu).

Chandrasekhar Potluri is with MCERC, School of Engineering, Idaho State University, Pocatello, Idaho 83209, USA (e-mail: potlchan@isu.edu).

Adnan Ilyas is with Measurement and Control Engineering Research Center (MCERC), School of Engineering, Idaho State University, Pocatello, Idaho 83209, USA (email: ilyaadna@isu.edu).

Parmod Kumar is with Measurement and Control Engineering Research Center (MCERC), School of Engineering, Idaho State University, Pocatello, Idaho 83209, USA (email: kumaparm@isu.edu).

Steve Chiu is with Department of Electrical Engineering and Computer Science, MCERC, Idaho State University, Pocatello, Idaho 83201 USA (email: chiustev@isu.edu).

Nancy Devine is with Department of Physical and Occupational Therapy, Idaho State University, Pocatello, Idaho 83209, USA (email:devinanac@isu.edu).

Alex Urfer is with Department of Physical and Occupational Therapy, Idaho State University, Pocatello, Idaho 83209, USA (email:urfealex@isu.edu).

Marco P. Schoen is with Department of Mechanical Engineering, MCERC, Idaho State University, Pocatello, Idaho 83209, USA (email: schomarc@isu.edu).

Hence, prosthetics, such as artificial hands, and their advances in functionality and performance will remain an active research topic. Functionality can be achieved by incorporating some kind of a sensory feedback, which allows the user to acquire external information through the artificial hand. Furthermore, the functionality of prosthesis can be improved by using a close to natural control mechanism. Research has shown that the adaptation of the prosthetic arm is conditional to a system that provides distally referred sensations of touch and joint movements [2]. It is postulated that to achieve greater dexterity and performance, prostheses without a sensory feedback system will be obsolete, [3-5]. It was assumed for a long time that the replacement of a natural arm with intelligent prosthetic arm is an impossible task. The main discrepancy lied in the insufficient analysis of the concept of sensory feedback and by not taking into account the knowledge of physiology of kinesthesis [6]. However with the passage of time, and as the technology matured, these factors were taken into account and later formed the base for a modern prosthetic hand. The sensory feedback has opened the new doors to the life of disable subjects compared to the earlier days. By using the sensory feedback, natural ambiences can be provided to the prosthetic hand users. It will help the user to have more control on the hand and can be fully embedded into the work as a normal person without any disability [7].

Sensory feedback can be obtained by the use of different implantable electrodes, such as needle or cuff electrodes. For robotic hands, [8] used this principle to excite the responsible nerves directly. They implanted the electrodes in fascicles of the nerves of amputees. The stimulation through these electrodes gives the feedback information on grip strength and position of the limb. This is not a closed-loop system and has some limitations in terms of optimization.

External sensors and switches are usually used in controlling functional neuromuscular simulation systems (FNS). They pose problems such as donning, and calibration. For implementation artificial sensors are difficult to build and are insufficiently bio compatible. Now-a-days with the advancement of electrical interfacing with nerves and muscles, natural sensors are being considered as an alternative source of feedback and command signals for FNS. For high-level control natural or artificial sensors can equally perform for decision making methods. Surface electromyography (sEMG) signals are being 1000 times larger than electro neuro organs are easier to measure, but have not provided reliable indicators so far. Characteristics like muscle fatigue are not indicated by these which are in FNS systems [9]. *Andrew Y.J. Szeto* [10] explained about the electrocutaneous stimulation for sensory communication in rehabilitation engineering.

Some procedures for implementing electro tactile displace and generating reliable, pain free sensations with a useful communication bandwidth. This paper presents an overview of what technologies have been developed in recent years and tries to forecast the direction of future research based on the current state of knowledge for advancing sensory feedback based prosthetic hands.

II BIOLOGICAL BACKGROUND

Some efforts have been made in recent years to address sensory feedback for the prosthetic users, [11]. The sensory feedback system interacting with the prosthetic and sensory cortex needs to be capable of communicating with both systems simultaneously, avoid issues of fatigue, assimilation, as well as be capable of sensing through the mechanical hand various types of signals, such as pressure, texture, temperature, shear forces, and surface conditions. In the following, we present a brief review of the inner workings of sensation through healthy human skin. There are numerous afferent receptors in the human skin to sense different sensation ranging from touch to temperature. In particular, there are cutaneous mechanoreceptors, thermoreceptors, nociceptors, bulboid corpuscles and chemoreceptors.

Cutaneous Mechanoreceptors are free nerve endings, sensing touch, pressure, and stretch. They are classified into four main types in the human skin:

1. Ruffini's end organs detect touch, pressure and tension deep in the skin. They are located all over the skin and are rather slowly adapting nerve endings and sensitive to skin stretch. Ruffini's end organs are mainly helpful with sense of and control of finger position and movement; they are also useful for detecting slippage of objects along the surface of the skin. Hence they help control grip force.
2. Meissner's corpuscle (or tactile corpuscle) is a mechanoreceptor (nerve ending). The location of Meissner's corpuscles is the glabrous skin. They detect changes in texture (most sensitive if vibration occurs below or around 50 Hz), are receptive to light touch sensation. Meissner's corpuscles are dynamic in nature, which results into rapid adaptation to external stimuli.
3. Pacinian corpuscles are fewer in number compared to Meissner and Merkel's discs, but adapt very rapidly. The location of Pacinian corpuscles are in subcutaneous tissue, interosseous membranes. Pacinian corpuscles detect rapid vibrations (about 200-300 Hz). They are nerve endings in the skin which are sensitive to pain and deep pressure (poking), and are also dynamic (adapt to stimuli).
4. Merkel's discs are located in all of the human skin and in hair follicles. Merkel's disc detects sustained touch and pressure and can distinguish shapes and textures. These receptors are good for touch and pressure. Their adaptation rate is rather slow as they are static in nature.

The location of the four mechanoreceptors is shown in Figure 1.

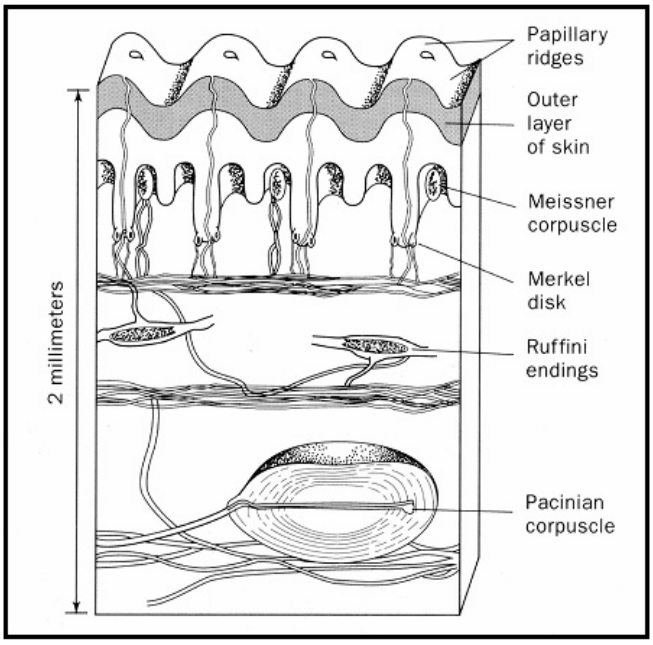


Figure 1: Mechanoreceptors location in human skin [12].

Thermoreceptors: are sensory neurons that sense changes in temperature. Heat is sensed through unmyelinated C-fibers, which possess a low conduction velocity. This result into the transmission of sensed information transmitted to the brain within a few seconds. Cold is sensed using C-fibers and thinly myelinated A-delta fibers, which conduct the information to the brain faster (within one second) [8]. For warm receptors, the warming effect is translated into an increase in action potential discharge rate, while cooling results in a decrease of the discharge rate. For cold receptors this is inverted, where the action potential firing rate is increased when cooling occurs and decreased when warming occurs. There is some literature [13]which notes that above 45^{o}C, some cold receptors also respond with an action potential discharge due to the increased temperature (paradoxical response to heat).

Cutaneous Nociceptors: is a skin receptor responsible of detecting damaging stimuli and alarming the system by creating the perception of pain. There are also thermal nociceptors which generate heat pain for temperatures above 42^{o}C.

While healthy subjects have access to all the information gathered by these receptors, individuals with amputations have to relearn how to collect information about an object they touch with the prosthetic. Recent research has shown that sensory feedback of contact information from the prosthetic with its surroundings improves the user's ability to control

and adapt to the prosthetic [14]. Currently, most work addressing the inclusion of feedback for the prosthetic user addresses only very basic elements such as tactile feedback about finger pressure.

III. SENSORY FEEDBACK TYPES

Holmes, [14] addressed the mental representation for sensory inputs of the human body within the brain. This input from the periphery has a strong effect on the perceptual awareness of body parts, natural or appended prosthesis for identifying its own body and for identifying artificial limbs. With sensory inputs, control and handling of the prostheses are improved. Some efforts have been made in recent years to address sensory feedback for the prosthetic users. For example, [15] used electrical stimulation to communicate pressure or position of the prosthesis in order to provide a closed- loop control of the hand. The most successful methods for communicating peripheral inputs were Electrical Surface Simulation (ESS) and Mechanical Surface Simulation (MSS), [16, 17]. Research shows that electro-tactile stimulators allow the distinction of 59 different sensations; the mechanical vibrator is able to transmit 16 different sensations, [18].

Tactile sensors can be split into two types: active and passive. In the general scenario, tactile sensory user interfaces passive touch i.e., stimulation is made passively on the surface of the skin. In some instances, in which when an object needs to be identified active touch comes into picture. This active touch utilizes distinct shapes and texture and encodes this sensory information which helps to identify the object without visual contact. In the active touch scenario shape encoding is very important, [19].

The tactile sensors can be used with electrical, pneumatic, and electro-mechanical devices. In the following sections the two most prevailing sensory feedback systems are discussed.

Mechanical Stimulation

In the case of electro-mechanical (vibro tactile) devices, a mechanical vibration or touch is produced and super-positioned onto a healthy skin area with functional mechano-receptors. The vibro tactile displays are classified into two basic types. They are pins and large point contact stimulators, [20].

The pin type vibro tactile displays based on piezoelectric bimorph pins are convenient and simple to use. They are non-invasive with good two point discrimination. These pin types are very good in presenting fine cues for surface texture edge and line detection when used as an array, [21]. In contrast, large contact point stimulators are very simple vibrators that are pressed against the skin surface. They yield much lower resolution when compared to the pin type. But the advantage with these types of simulators is that they can be distributed over large section of the body. Therefore multiple simultaneous cues for surface texture, edge and line detections [22] can be produced.

The mechanical stimulation uses pneumatic devices such as bladders or pockets which can be inflated or deflated, [23]. Hence they create a pulsing sensation that the user can easily feel. These devices can either be attached directly or to other devices that are used to complete the tasks. The main advantages of these pneumatic devices (bladders) – when compared to vibro tactile displays – are localization i.e., they have significantly low interference with the nearby bladders, the pump mechanism can be mounted remotely such that the control devices will require a minor modification. A variety of sensory information of the stimulus can be generated by altering the configuration or shape of these bladders [23, 24].

A number of studies have been conducted using the mechanical vibrator as basis [11, 15, 25, and 26]. This approach generally leads to a higher acceptance rate by the user as well as an increased users' performance (time and efficiency of use). However, there are some limitations with mechanical stimulation, mechanical vibration is limited to 16 sensations when used as a source of sensory feedback, they are bulkier and hard to control, and the pneumatic tactile displays include leak control, and air compressibility issues.

Electrical stimulation

In order to design better prostheses with vibro-tactile feedback, electrical stimulation of the muscles is necessary. By applying the electrical currents paralyzed muscles can be made to contract. These electrically elicited muscle contractions are controlled in such a way that the prosthesis acquires its full functionality. This technique is called "functional electrical stimulation". There is a lot of ongoing research on FES systems to restore the functionality of the upper and lower extremity prosthesis [27]. *Kajimoto et. al,*[28] used an electrode array with a particular distribution of the electrode's charge to communicate with the user, the sensation of pressure at the fingertips. Their smart touch prototype consists of optical sensors to capture an image and convert that into tactile information and displays through electrical stimulation. Using an electro-tactile system makes it possible to access seven (7) different classes of mechanoreceptors, two (2) classes of thermo receptors, four (4) classes of nocioceptors, and three (3) classes of proprioceptors within the human skin, [28, 29, 30, 31, 32, 33]. To utilize the possible 59 sensations from electrical stimulations, Szeto et. al [32] used electro-tactile stimulations by changing frequency and intensity of the feedback. In [34], the authors used interferential stimulation to transit spatial movement of the hand. Electrical stimulation is described in literature by using voltage-regulated stimulation, [34]. Voltage-regulated systems are sensitive to changes in the amputee's skin impedance. In reference [35], this problem was addressed by creating a voltage-regulated stimulator with a high frequency biphasic waveform. This approach lead to

the avoidance of sensitivity (ability to discriminate objects) drop off. Array sensors have been used successfully for different purposes [36, 37]. As above mentioned, work presented in [28] utilize an electrode array. A rather large array was used by [38], where 144 electrodes are used on a blind person's tongue in order to provide some spatial awareness of his/her surroundings. Electrode arrays for feedback have numerous advantages and can help tailor the sensory feedback system to the skin area chosen.

IV CONCLUSION AND FUTURE WORK

In this paper, we addressed issues that concern the integration of artificial limbs t the amputees. There are two systems of note that can be used to structure an afferent system. The two systems are either based on vibration or on electrical stimuli or a combination of the two methods. Electrical Stimulation provides the better performance compared to the mechanical stimulation, measured by the type and variety of possible sensations generated and perceived by the mechanoreceptors in the human skin. Designing a prosthetic hand with a sensory feedback technique will increase the acceptability by the user of the prosthetic hand and hence the control of the hand [39, 40].

ACKNOWLEDGEMENT

This work was supported in part by an internal Career Path Internship grant by Idaho State University. The support is greatly appreciated. The technical support from Dr. D. Subbaram Naidu is greatly appreciated.

REFERENCES

[1] Estimating the Prevalence of Limb Loss in the United States- 2005 to 2050 Kathryn Ziegler-Graham, PhD, et al, *Archives of PhysicalMedicineand Rehabilitation* 89(3):422-429, 2008.

[2]http://content.lib.utah.edu/cdm4/document.php?CISOROOT=/ir-main&CISOPTR=43804.

[3] J. N. Billock, "*Prosthetic management of complete hand and arm deficiencies*," in:J.M. Hunter, E.J. MacKin and A.D. Callahan(Eds.),*Rehabilitation of the Hand: Surgery and Therapy*, St. Louis: Mosby-Year Book,1189-1201, 1995.

[4] S. Hubbard, "*Myoprosthetic management of the upper limb amputee*," in:J.M. Hunter, E.J. MacKin andA.D. Callahan(Eds.),*Rehabilitation of the Hand: Surgery and Therapy*, St. Louis: Mosby-Year Book,1241-1252, 1995.

[5] G. Lundborg, B. Rosen, K. Lindstrom and S. Lindberg, "*Artificial sensibility based on the use of piezoresistive sensors:preliminary observations*," *J.Hand Surg.Br.*, 23:620-626, 1998.

[6] P. Herberts, L. Korner, "*Ideas on Sensory feedback in hand prostheses*", Prosthet Orthot Int, December 1979, vol 3. No. 3 157-162.

[7]http://www.research.gov/researchportal/appmanager/base/desktop;jsessionid=5JSnNMdMdnn5n4vhVrCp32VWlvjJnLsmfdc19DcYbkJd5jrwSQMt!1821093771!1028386895?_nfpb=true&_windowLabel=awardSummary_1&_urlType=action&awardSummary_1_action=selectAwardDetail&awardSummary_1_id=%2FresearchGov%2FAwardHighlight%2FPublicAffairs%2F16782_SensoryFeedbackfromaProstheticHand.html

[8] Christian Antfolk, Christian Balkenius, et al., "A *tactile Display system for Hand prostheses to discriminate pressure and individual finger localization*"

[9] Dhillon, G. S., and Horch K. W., IEEE Trans. Neural Systems and Rehabilitation Engineering, "*Direct neural sensory feedback and control of a prosthetic arm*," 13(4), pp. 468-472, 2005.

[10]Szeto, A. Y. J., Riso, R., 1990, "*Sensory Feedback using Electrical Stimulation of the Tactile Sense*", in R.V. Smith and J.H. Leslie, Jr. (Eds.), *Rehabilitation Engineering*, Boca Raton, CRC Press, pp. 29-78.

[11] Cipriani, C. Zaccone, F., Micera. S., Carrozza, M. C., 2008, "*On the Shared Control of an EMG Controlled Prosthetic Hand: Analysis of User-Prosthetics Interaction*," IEEE Trans. Robotics, 24(1), pp. 170-184.

[12]http://www.sensorprod.com/news/whitepapers/2006_tsa/index.php

[13]Darian-Smith, Ian; Johnson KO, LaMotte C, Shigenaga Y, Kenins P, Champness P (1979). "*Warm fibers innervating palmar and digital skin of the monkey: responses to thermal stimuli*.".*Journal of Neurophysiology*42 (5): 1297–1315.

[14] Holmes, N. P., Spence, C., Cogn. Process,"*The body schema and multisensory representation(s) of peripersonal space*," 5, 94-105, 2004.

[15] Pylatiuk, C., Kargov, A., Schulz, S., Journal of Prosthetics and Orthotics, "*Design and Evaluation of a Low-Cost Force Feedback System for Myoelectric Prosthetic Hands*," 18(2), pp. 57-61. 2006a.

[16] G.F. Shannon, "*A comparison of alternative means of providing sensory feedback on upper limbProstheses* ", Med Biol Eng. 1976 May; 14(3), 289-294.

[17] Geldard, F.A. (1957) Adventures in tactile literacy. Am psycho 115-24.

[18]Kaczmarek, K. A., Webster, J. G., Bach-y-Rita, P., Tompkins, W. J., IEEE Trans. Biomedical Engineering,"*Electro-tictile and vibrotactile display for sensory substitution systems*," 38, pp. 1-15, 1991.

[19]M. Akamatsu, I.S. MacKenzie, and T. Hasbrouc, "*A Comparison of Tactile, Auditory, and Visual Feedback in a Pointing Task Using a Mouse-Type Devic*e," Ergonomics 38, pp. 816-827, 1995.

[20] S. Brewster and L.M. Brown, "*Tactons: Structured Tactile Messages for Non-Visual Information Display*," Proc. 5th Australasian User Interface Conference (AUIC2004), Dunedin, pp. 15-23, 2004.

[21] J.C. Bliss, M.H. Katcher, C.H. Rogers, R.P. Sheppard, "*Optical-to-tactile image conversion for the blind*," IEEE Trans. Man-Machine Systems, MMS-11(1), pp. 58-65, 1970.
[22]M. Schrope, "*Simply Sensational*," New Scientist, pp. 30-33, June 2, 2001.
[23]M. Enriquez, O. Afonin, B. Yager, and K. Maclean, "*A Pneumatic Tactile Alerting System for the Driving Environment*," PUI 2001, Workshop on Perceptual/Perceptive User Interfaces, Orlando, FL, November 15-16, 2001.
[24] Lilly Spirkovska, "Summary on Tactile Interfaces techniques and Systems, NASA Ames Research Center, September 22, 2004.
[25] Pylatiuk, C., Doederlein, L, Orthopade, "*Bionic Arm prostheses. State of the art in research and development*,", 35(11), pp. 1169-1170, 2006b.
[26] Schulz, A. E., Marasco, P. D., Kuiken, T. A., Brain Research, "*Vibrotactile detection thresholds for chest skin of amputees following targeted reinnervation surgery*,", 1251, pp. 121-129, 2008.
[27] P. Hunter pecham, Jayme S. Knutson, "*Functional electrical simulation for neuromuscular applications*", Annu. Rev. Biomed.Eng, March 23, 2005.
[28] Kajimoto, H., Kawakami, N., Maeda, T., Inami, M., IEEE Computer Graphics and Applications," Emerging Technologies, "*SmartTouch: Electric Skin to Touch the Untouchable*", 24(1), pp. 36-43, 2004.
[29] Kaczmarek, K. A., Webster, J. G., Bach-y-Rita, P., Tompkins, W. J., IEEE Trans. Biomedical Engineering,"*Electro-tictile and vibrotactile display for sensory substitution systems*," 38, pp. 1-15, 1991.
[30] Kajimoto, H., Kawakami, N., Maeda, T., Tachi, S., Electronics and Communications in Japan, "*Electrocutaneous Display with Receptor Selective Stimulations*", Part 2, 85(6), pp. 40-49, 2002.
[31] Kajimoto, H., Kawakami, N., Inami, M., Tachi, S., Proceedings of the Annual Conference on Artificial Reality and Tele existence ICAT 99, Virtual Reality Soc. of Japan, "*Tactile feeling Display using Functional Electrical Stimulation*" pp. 107-114, 1999.
[32] Szeto, A. Y. J., Riso, R., in R.V. Smith and J.H. Leslie, Jr. (Eds.), Rehabilitation Engineering, Boca Raton, CRC Press, "*Sensory Feedback using Electrical Stimulation of the Tactile Sense*", pp. 29-78, 1990.
[33] Asamura, N., Yokoyama, N., Shinoda, H., Proceedings IEEE Virtual Reality Conference, "*A Method of Selective Stimulation to Epidermal Skin Receptrots for Realistic Touch Feedback*", pp. 274-281, 1999.
[34] Navarro, X., Krueger, T., Lago, N., Micera, S., Stieglitz, T., Dario, P., Journal of the Peripheral Nervous System, "*A critical review of interfaces with the peripheral nervous system for the control of neuroprostheses and hybrid bionic systems*", 10(3), pp. 229-258. 2005.
[35] Hernandez, A., Yokoi, H., Ohnishi, T., a Arai, T., Proceedings of the 9th Int. Conf. On Intelligent Autonomous Systems, IOS Press, Tokyo, Japan, "*An f-MRI study of an EMG prosthetic hand biofeedback system*", pp. 921-929, 2006.
[36] Chandrasekhar Potluri, Parmod Kumar, Madhavi Anugolu, Alex Urfer, Steve Chiu, D. Subbaram Naidu, and Marco P. Schoen, "*Frequency Domain Surface EMG Sensor Fusion for Estimating Finger Forces,"* 32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Buenos Aires, Argentina, Aug. 31 – Sept. 4, 2010.
[37] Madhavi Anugolu, Anish Sebastian, Parmod Kumar, Marco P. Schoen, Alex Urfer, and D. Subbaram Naidu, "*Surface EMG Array Sensor Based Model Fusion using Bayesian Approaches for Prosthetic Hands*," Proceedings of the Dynamic Systems and Control Conference (DSCC), Hollywood, CA, October 2009.
[38] Paul Bach-y-Rita, Stephen W. Kercel, Sensort substitution and the human-machine interface, Trends in Cognitive Sciences, Volume 7, Issue 12, December 2003, Pages 541-546.
[39] Chandrasekhar Potluri, Parmod Kumar, Madhavi Anugolu, Steve Chiu, Alex Urfer, Marco P. Schoen, and D. Subbaram Naidu, "*sEMG Based Fuzzy Control Strategy with ANFIS Path Planning For Prosthetic Hand*," 3rd IEEE RAS &EMBS International Conference on Biomedical Robotics and Biomechatronics, Tokyo, Sept 26-30, 2010
[40] Chandrasekhar Potluri, Yimesker Yihun, Parmod Kumar, Jeff Molitor, Steve C. Chiu,D. Subbaram Naidu, *Fellow, IEEE*, S.Hossein Mousavinezhad, "*sEMG Based Real-Time Embedded Force Control Strategy for a Prosthetic Hand Prototype*" IEEE International Conference on Electro/Information Technology, Mankato, Minnesota, USA, May 15-17, 2011.

An Activity Recording System with a Radial-Basis-Function-Network-Based Energy Expenditure Regression Algorithm

Jeen-Shing Wang[1], Ya-Ting Yang[2], Che-Wei Lin[1]
[1]Department of Electrical Engineering,
[2]Institute of Education & Center of Teacher Eduction
National Cheng Kung University
Tainan 701, Taiwan, R.O.C.
jeenshin@mail.ncku.edu.tw

Abstract - *This paper presents an activity recording (AR) system and a radial-basis-function-network-based (RBFNB) energy expenditure regression algorithm. The AR system includes motion sensors and an electrocardiogram sensor which is composed of a set of sensor modules (accelerometers and electrocardiogram amplifying/filtering circuits), a MCU module (microcontroller), a wireless communication module (a RF transceiver and a Bluetooth® module), and a storage module (flash memory). A RBFNB energy expenditure regression algorithm consisting of the procedures of data collection, data preprocessing, feature selection, and construction of energy expenditure regression model, has been developed for constructing energy expenditure regression models. The sequential forward search and the sequential backward search were employed as the feature selection strategies, and a radial basis function network as the energy expenditure regression model in this study. Our experimental results exhibited that the proposed energy expenditure regression algorithm can achieve satisfactory energy expenditure estimation by combing appropriate feature selection technique with the regression models.*

Keywords: Energy expenditure, accelerometer, electrocardiogram, feature selection, and radial basis function network.

1 Introduction

Inactive lifestyle is a key factor which causes chronic diseases such as diabetes, obesity, and cardiovascular diseases [1]. In order to improve people's lifestyle from inactive to active, providing necessary information such as energy expenditure of daily activities is essential. In the past decades, owing to the rapid development of MEMS and IC technology, physiological signal sensors such as accelerometers for motion detection and electrocardiogram (ECG) for cardiovascular diseases became cheaper and easier to use. Much literature aimed at using activity acceleration or heart rate data to develop energy expenditure regression algorithms [2],[3],[4],[5].

This paper proposed an AR system and an energy expenditure regression algorithm. The AR system can be used in the collection of acceleration signals and ECG data simultaneously. The acceleration signal and ECG data can be used for constructing an effective regression model by the proposed RBFNB energy expenditure regression algorithm. The proposed regression algorithm includes data collection, data preprocessing, feature selection, and construction of energy expenditure regression model. In order to construct a regression model with the least number of features and good regression performance, the sequential forward search (SFS) and the sequential backward search (SBS) [6] were employed as the feature selection techniques, and a radial basis function network (RBFN) was employed as the regression model of energy expenditure. The effectiveness of the proposed energy expenditure regression algorithm was evaluated by a RBFN-based model for energy expenditure estimation using a set of laboratory-controlled activities. The advantage of the propose energy expenditure regression algorithm is to achieve a satisfactory energy expenditure estimation with the least number of features.

2 AR System

We have developed an AR system which is composed of motion sensors and an ECG sensor. The motion sensors are mounted on users' wrist, waist, and ankle and ECG sensor is attached on users' breast. The motion sensors are responsible for sensing acceleration signal from users' daily activities. The ECG sensor is responsible to collect users' ECG data during activities. The hardware components of each sensor include a signal-sensing module, a MCU module, a wireless communication module, and a storage module. The major component in the motion sensor is a triaxial digital accelerometer for sensing motion/activity acceleration signal and the sensing module in the ECG sensor is an amplifying/filtering circuit. The accelerometer IC employed in

the AR system is Freescale® MMA7455L triaxial digital output accelerometer. The Freescale® MMA7455L accelerometer possesses a user selectable full scale of ±2g, ±4g, and ±8g and is able to measure acceleration signal over the bandwidth of 125 kHz for all axes. The sensitivity of accelerometers was set from -8g to 8g in this study. The ECG amplifying/filtering circuit is composed of an instrument amplifier, a high-pass filter, a low-pass filter, and a notch filter. The instrument amplifier is responsible for amplifying differential input signals from the heart of a user, and rejecting common-mode noise such as electromyography (EMG) interference in ECG. The common mode rejection ratio (CMRR) of the instrument amplifier is higher than 90 dB in this study. The high-pass, low-pass, and notch filters in our design are responsible to reject the noises that are out of ECG's signal bandwidth. According to the nature of ECG, the filters are set to attenuate the signal that is under 0.05 Hz (by the high-pass filter), above 100 Hz (by the low-pass filter), and 60 Hz (power-line interference) (by the notch filter).

The Microchip® PIC24FJ64GA002 was selected as the MCU module. The MCU module is responsible for the following tasks: 1) timing flow control (regularly retrieving data from the sensor module via I2C bus from accelerometers or from the 12-bit A/D converter embedded in the MCU from the ECG sensor), 2) wireless communication control, 3) peripheral component control. The wireless communication module includes a Nordic® nRF24L01+ wireless RF transceiver and a BTM-162 Bluetooth® module. The Nordic® nRF24L01+ wireless RF transceiver is employed to execute timing synchronizations and data transmission from the motion sensors and ECG sensor. The BTM-162 Bluetooth® module is served as the standard communication channel between the AR system and standard devices such as PCs or smart phones. The storage module employed in the AR system is MXIC® MX25L128 flash memory with 16 MB storage capacity. Considering high performance and low power consumption, we selected the Microchip® PIC24FJ64GA002 as the main controller. The sampling rates (f_s) of the motion sensors and ECG sensor are 30 Hz and 200 Hz with 8-bit and 12-bit data resolution, respectively. The power consumption of motion sensors and ECG sensor is 12.2 mA and 20 mA, respectively. The sizes of the motion sensors and the ECG sensor are 32mm × 30mm × 5mm and 35mm × 19mm × 6mm, respectively.

3 RBFNB Energy Expenditure Regression Algorithm

The proposed RBFNB energy expenditure regression algorithm includes the procedures of data collection, data preprocessing, feature selection, and construction of energy expenditure regression model as shown in Fig. 1. At the beginning, the data collection process collects users' activity acceleration signal and ECG data by the AR system. The collected data is stored in the memory module of the AR system, and can transmit to a PC via the Bluetooth® communication protocol. The processes of data preprocessing, feature selection and energy expenditure regression model construction are all executed on a PC.

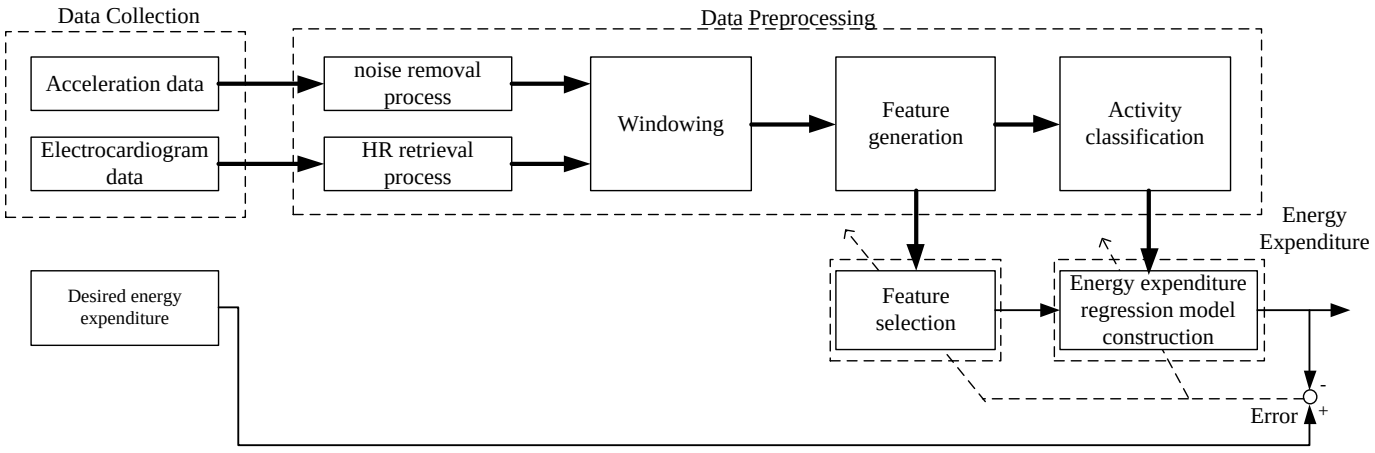


Fig. 1. The flowchart of the RBFNB energy expenditure regression algorithm.

The data preprocessing process includes a noise removal process for acceleration signal, heart rate (HR) retrieval process for ECG signals, a windowing process, a feature generation process, and an activity type classification process. The purpose of noise removal process is to reduce the effect of baseline drift and high frequency noise from the collected acceleration signal. The HR retrieval process is to obtain the HR series from ECG in order to extract important features for activity type classification and energy expenditure regression model construction. The windowing process is to segment the

acceleration signal and HR data into consecutive windows for feature generation. After the windowing process, the feature generation process generates statistical features from the acceleration signal and HR data for activity classification and energy expenditure regression model construction. The purpose for activity classification is to construct energy expenditure regression models for different types of activities for better energy expenditure estimation.

After the data preprocessing process, SFS and SBS are used in the feature selection process to select significant features for constructing energy expenditure regression models efficiently. Finally, an RBFN was trained as the energy expenditure regression model.

4 Experimental Design and Results

The effectiveness of the AR system and the accuracy of the proposed RBFNB energy expenditure regression algorithm were examined by the following experimental setup. We recruited 26 college students and each participant was asked to perform fourteen activities listed in Table 1. During the experiment, all participants wore a gas analyzer (K4b^2) and the AR system. The gas analyzer was used to collect the participants' energy expenditure during activities while the AR system was used for collecting participants' acceleration signals during activities.

Table 1. Types of activities and the corresponding average HR and MET

Type	Activity	Average HR	Average MET
1	Lying	72.31	1.28
	Desk working	81.74	1.49
	Sitting	82.03	1.41
	Standing	90.43	1.45
2	Sweep	96.40	3.18
	Mopping	99.63	3.44
	Walking (3mph)	97.77	3.75
	Walking (4mph)	101.98	4.37
	Downstairs	115.43	3.68
	Bicycling (50Watt)	125.83	5.59
3	Upstairs	136.67	6.65
	Running (6mph)	137.90	8.49
	Running (7mph)	151.78	9.41
	Bicycling (100Watt)	151.88	8.26

Twenty-three features (12 features derived from the acceleration signals, 9 features derived from the HR data, and 2 personal information features including weight and height) were generated as the feature candidates of the energy expenditure regression models. SFS and SBS were employed as the search strategies, and the reciprocal of MSE of the constructed energy expenditure regression model (RBFN/GRNN) was chosen as the criterion function. The numbers of selected features in the SFS/SBS process were set from one to seven. This setting was used to examine the relations between the number of features and the performance of different energy expenditure regression models.

After the feature selection process, the performance of energy expenditure regression models was evaluated by constructing an RBFN using the selected features. The performance was evaluated by the standard error of the estimation (SEE) and the coefficient of determination (R^2) between the desired energy expenditure and the estimated energy expenditure. The SEE and R2 of RBFN with the top 7 features are shown in Fig. 2 and Fig. 3. From the observation of Fig. 2 and Fig. 3, the performance of the SBS is better than that of SFS in the evaluation of SEE and R2. The SEE and R2 in the RBFN regression model with the top 7 features can achieve 0.24/0.45/0.64 (SEE) and 0.39/0.82/0.87 (R^2) for Type 1, 2, and 3, respectively.

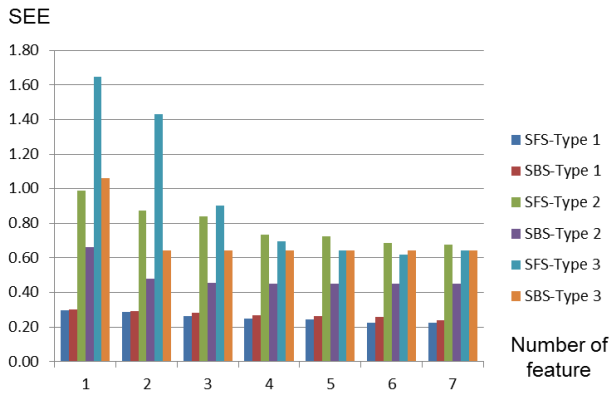


Fig. 2. SEE of energy expenditure regression model based on SFS/SBS and RBFN as the criterion function.

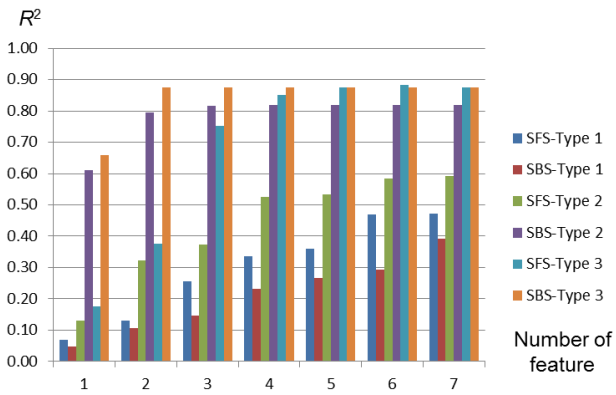


Fig. 3. R^2 of energy expenditure regression model based on SFS/SBS and RBFN as the criterion function.

5 Conclusion

A realization of the AR system and the development of a RBFNB energy expenditure regression algorithm have been presented in this paper. The AR system can be used to collect the acceleration signals of daily activities and ECG data simultaneously. In order to construct an efficient regression model with the least number of features and good regression performance, SFS and SBS were employed as the feature selection techniques. From these results, we have successfully validated the effectiveness of the proposed system as well as the construction algorithm.

6 References

[1] Physical Inactivity and Cardiovascular Disease, http://www.health.state.ny.us/diseases/chronic/cvd.htm

[2] M. P. Rothney, E. V. Schaefer, M. M. Neumann, L. Choi, K. Y. Chen, "Validity of Physical Activity Intensity Predictions by Actigraph, Actical, and RT3 Accelerometers"; Obesity Society, vol. 16, 1946—1952, 2008

[3] S. E. Crouter, J. R. Churilla, Jr, D. R. Bassett, "Estimating Energy Expenditure using Accelerometers"; Eur. Journal of Applied Physiology, vol. 98, 601—612, 2006

[4] Jr, D. R. Bassett, B. E. Ainsworth, A. M. Swartz,, S. J. Strath, W. L. O'brien, G. A. , King, "Validity of Four Motion Sensors in Measuring Moderate Intensity Physical Activity"; Medicine & Science in Sports & Exercise, vol. 32, 471—480, 2000

[5] K. Y. Chen, M. Sun, "Improving Energy Expenditure Estimation by using a Triaxial Accelerometer"; J Appl. Physiol, vol. 83, 2112—2122, 1997

[6] P. Pudil, J. Novovičová1, and J. Kittlera, "Floating Search Methods in Feature Selection"; Pattern Recognition Letters, vol. 15, 1119—1125, 1994

USING SEARCH AND LEARNING FOR PRODUCTION OF RESOURCES IN RTS GAMES

Augusto A. B. Branquinho, Carlos R. Lopes, Thiago F. Naves
Faculty of Computing, Federal University of Uberlândia, Uberlândia, Minas Gerais, Brazil

Abstract— *So-called real-time strategy games are characterized by two important steps. In the first step a plan of action should be carried out to produce resources. In the second step, the resources produced in the former step are employed in battles against the enemy. Resource production is vital to succeed in this sort of game. This paper describes an algorithm for planning and scheduling actions that provide the gathering of resources. Emphasis in this paper is given to the scheduling strategy, which is based on SLA*. In order to increase the efficiency of the scheduling, some strategies were developed to improve SLA*. Experiments show better results compared to other existing approaches.*

Keywords: Real-Time Strategy Games, Resources, Planning, Search, Learning

1. Introduction

Real Time Strategy (RTS) games is one of the most popular categories developed in the universe of games. Recent titles such as *Starcraft II* and the world renowned *World of Warcraft* are examples of this category. RTS games are mostly characterized by battles and wars against enemies that can be performed by another human player or a computer.

It is possible to identify two phases in a RTS game. First, there is an initial period in which each player starts the game with some units and/or buildings and develops his army via resource production. In the next phase military campaigns take place. In this phase the resources gathered earlier are employed for offense and defense. Therefore, the stage of resource production is vital to succeed in this game.

The resources in RTS games are all kinds of raw materials, basic construction, military units and civilization. For obtaining a desired set of resources (a goal to be achieved) it is necessary to carry out actions. In general we have two sorts of actions related to resources: actions that produce resources and actions that collect resources. Once we have actions (a plan) that achieve a goal it is necessary to schedule them. As a result we can have a gain in performance because some actions can be run in parallel with others. This gain in performance is achieved by a **makespan** reduction, which is one of the main goals of our work.

With respect to the task of generating a plan of action to achieve a set of resources, we developed an algorithm, which is called **MEAPOP**. Our work was built up on the approach developed by Chan [1] who developed an algorithm for resource production based on **MEA** [1]. However, rather than generating a linear sequence of actions, MEAPOP produces a **partial order plan** of actions to achieve a resource goal. MEAPOP is described in details in [2].

In this paper, we focus on the task of scheduling the actions that make up the plan. To do this, we make use of **SLA*** [3], an algorithm based on search technique and learning for real-time purposes. Our work is motivated by the results obtained in [1] and by an approach that also uses SLA* for scheduling activities that work with a fixed number of resources [4]. However, in RTS games the number of resources is dynamic. To take into account this feature we adapted and developed strategies to improve the efficiency of the algorithm, which are the main contributions described in this paper. Such strategies are based on elaborate data structures and careful choice of states and actions to be used in the search process. Their employment allows us to achieve better results in terms of makespan reduction in comparison to other approaches.

We understand that this research is interesting for the AI planning field because it deals with a number of challenging problems such as actions with numeric effects, concurrent activities, and real-time constraints. As a matter of fact, some testbeds have been developed based on RTS games [5]. Also, the mechanism developed can be useful as an interface to assist a human player. The human player just specifies a resource goal and our component figures a plan of action that the player has to carry out to achieve the goal.

The remainder of the paper is organized as follows. Section 2 presents the characterization of the problem. Section 3 describes the partial order planning. The scheduling algorithm based on SLA* is presented in Section 4. In section 5 we describe the strategies developed for improving the scheduling. Details of the implementation are described in section 6. Results are presented in section 7 and a conclusion appears in section 8.

2. Characterization of the problem

The problem of planning in RTS games is to determine a sequence of actions to reach a goal state, in which a set of resources is achieved. This process should be efficient. In general some actions can be executed concurrently and make use of limited resources. This implies that some scheduling should be done in order to minimize the time to achieve the

resource production goal, which is known as makespan. In this way, the whole planning process has to come up with actions and their scheduling to achieve a resource goal.

The resources to be produced and used fall into one of the following categories: *Require*, *Borrow*, *Produce* and *Consume*. The domain set of actions and resources used in the context of this work is shown in Figure 1, and these are based on the game Warcraft II.

```
resource gold
resource wood
resource supply
resource townhall
resource barracks
resource peasant
resource footman

action collect-gold :duration 510
:require 1 townhall :borrow 1 peasant
:produce 100 gold
action collect-wood :duration 1570
:require 1 townhall :borrow 1 peasant
:produce 100 wood
action build-supply :duration 620
:borrow 1 peasant :consume 500 gold 250 wood
:produce 4 supply
action build-townhall :duration 1530
:borrow 1 peasant :consume 1200 gold 800 wood
:produce 1 townhall
action build-barracks :duration 1240
:borrow 1 peasant :consume 700 gold 450 wood
:produce 1 barracks
action build-peasant :duration 225
:borrow 1 townhall :consume 400 gold 1 supply
:produce 1 peasant
action build-footman :duration 200
:borrow 1 barracks :consume 600 gold 1 supply
:produce 1 footman
```

Fig. 1: Specification domain set [6].

In Figure 1 the action *collect-gold* can be performed only if a *1 peasant* and a *1 townhall* are available, the peasant is a resource of type *Borrow*. This means that while the action *collect-gold* is being executed, the resources used by it are not available for other actions; only after its completion they become available for other actions. This situation well describes the challenges surrounding the scheduling of actions.

In order to derive actions that achieve a set of resources, existing approaches use algorithms that employ MEA [6]. The actions achieved are scheduled using a heuristic scheduler, which is not capable of finding the best solution for most of the problems. To illustrate, consider the following plan of action based on those described in Figure 1: *collect-gold*, *collect-gold*, *collect-gold* and *collect-wood*. Based on an initial state with a *(1 townhall, 2 peasant)* the following scheduling is generated for this goal state: two actions *collect-gold* in parallel followed by a single action *collect-gold* followed by another action to *collect-wood*. This scheduling produces a makespan with duration 2080. However, the best scheduling possible for that entry would be the action of *collect-wood* parallel with three actions of *collect-gold* achieving a makespan with duration 1570.

An example that shows the effectiveness of our approach for scheduling can be seen in Figure 2. In Figure 2 there are four specific actions that can be employed to reach the goal state (*1 townhall, 3 peasant*) from an initial state (*1 townhall, 1 peasant, 2 supply, 600 gold*). The plan described in Figure 2 A was obtained through the algorithm proposed in [6] and transformed into that which appears in Figure 2 B using the scheduler proposed in [6], which produces a makespan with value 1245. The plan in Figure 2 C was achieved using a partial order planner [2] and transformed into the plan depicted in Figure 2 D using the scheduler described in this paper. The result is a scheduling with a makespan with a duration 960, which in this case represents the optimal solution to this problem. Note that the order required for the actions in Figure 2 C was different from that obtained in Figure 2 A. This result is related to strategies that we developed in order to improve the performance of the scheduler using SLA*.

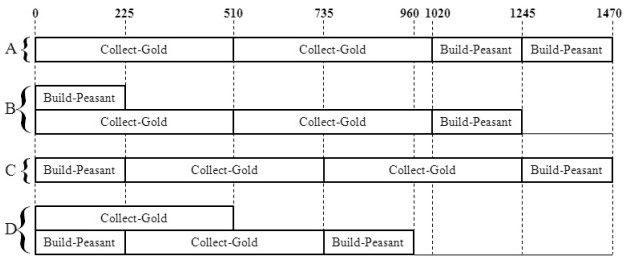


Fig. 2: Example of plans.

Comparisons of results presented above are motivators for this study since the existing approaches do not aim to obtain the best makespan. Our goal is to find the best solution with a response time compatible with the characteristics and challenges of scheduling in real-time environments, such as those described in this section, by using an algorithm of search and learning adapted for this task.

3. Partial Order Planning

For a better understanding of this work it is necessary to present some features of the algorithm responsible for creating the partial order plans, which is called MeaPop [2]. MeaPop generates a plan that serves as input for the scheduler based on SLA*. This plan is treated as a project of activities, i.e, a set of activities linked by precedence constraints. Figure 3 shows a partial order plan used to illustrate this process. A plan should be considered as a graph, where each vertex corresponds to an activity. Each activity is characterized by the following attributes: a quantity of actions, a number that represents it (an identification), the

largest amount of time to carry them out and a priority for each action (see Figure 3).

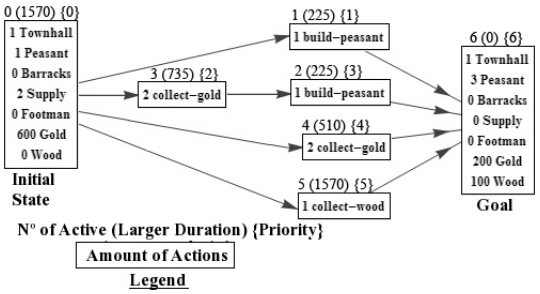


Fig. 3: Example of partial order plan to be used in scheduling.

The value of longer duration of activities is treated as the initial heuristic, used in the learning process of SLA*. A heuristic is a lower limit of the estimated duration for completing the remaining activities. It is important to note that every action has a duration, which is used in the estimation of heuristics. Activity 0 corresponds to the initial state of resources and activity 6 represents the goal to be achieved. The goal represents the minimum resources to be achieved, which were established by the goal state. Hence, there may be extra resources after the execution of all actions.

The priority information that is calculated in partial order planning does not define the order in which the actions of each activity should be scheduled; this only indicates a preference over lower priority. The following topics describe the scheduling process and the implementation details.

4. SLA* Applied in Scheduling Resources in RTS Games

SLA* is an algorithm for search and learning, which is considered a refinement of LRTA* [3]. It is capable of finding the optimal solution in just one iteration. As shown in Figure 3 the partial order plan handles seven resources and consists of a number of actions required to get the resources specified in the goal state, which makes the scheduling problem more difficult and time consuming. An important detail is that one action can influence the performance of others. For instance, actions that require the resource *gold* can only be executed when this resource is available. This might become possible only when actions that *collect-gold* produce the quantity required.

Regarding the characteristics presented above, it is important to note that each state of the graph corresponds to a scheduling of actions. As the activities of the plan are made by one or more actions, the algorithm checks for all possible distinct combinations that can be used in a given time. The scheduling considers the activities individually. Algorithm 1 shows the pseudo-code of the SLA* algorithm.

Algorithm 1 SLA*($s_{initial}, s_{goal}$)

1: $STACK.add(s_{current})$
2: **loop**
3: $s_{current} \leftarrow STACK.top()$
4: **if** $s_{current} = s_{goal}$ **then**
5: **return** $STACK$
6: **end if**
7: $s' \leftarrow arg\ min_{s'' \in succ(s_{current})}(k(s_{current}, s'') + h(s''))$
8: **if** $h(s_{current}) \geq k(s_{current}, s') + h(s')$ **then**
9: $STACK.add(s')$
10: **else**
11: $h(s_{current}) \leftarrow k(s_{current}, s') + h(s')$
12: **if** $s_{current} \neq s_{initial}$ **then**
13: $STACK.remove()$
14: **end if**
15: **end if**
16: **end loop**

According to the concepts already presented, each state consists of a set of actions that are in progress, i.e, actions being scheduled. For scheduling in RTS games, we developed an algorithm to find the successor states (line 7 of Algorithm 1), named *SucessorStates*.

As stated before, Algorithm 2, *SucessorStates(Resources, Activities)*, is responsible for creating successor states. The result of this procedure corresponds to the scheduling of actions present in *Activities* for the whole set of available resources in *Resources*. The parameter *Resources* represents the available resources and parameter *Activities* refers to activities that can be scheduled. Each activity consists of one or more actions of the same type. For example, activity 3 in Figure 3 has two actions *collect-gold*. Once a state is created, it is added to the variable *States*. The variable *S* corresponds to the actions that are being scheduled in a given time.

Algorithm 2 SucessorStates($Resources$, $Activities$)

1: Global variable $States \leftarrow \emptyset$, initially without elements. Successors will be created considering the resources $Resources$ and activities $Activities$.
2: $n \leftarrow$ quantity of activities in $Activities$
3: **for** $i \leftarrow 1, ..., n$ **do**
4: $S_i \leftarrow 0$
5: **end for**
6: `AddStates(`A_1`, S,` $Resources$`,` $Activities$`)`
7: **return** $States$

Algorithm 3 *AddStates (Aj, S, Resources, Activities)* is responsible for creating the states and adding them to the variable *States*. The first parameter specifies the initial activity of the scheduling. The S parameter shows the number of actions related to each activity that were scheduled in

a previous state. The resources available in $Resources$ are those considered for the execution of actions, i.e, an action can be part of a scheduling generating a new state only if these resources satisfy the preconditions of the action.

Algorithm 3 AddStates($A_j, S, Resources, Activities$)

1: **for each** Active $A_i \in Activities$, such that $i \geq j$ **do**
2: $\beta \leftarrow$ action associated with activity A_i
3: $s_i \leftarrow$ number of actions the activity A_i already scheduled in S
4: $a_i \leftarrow$ number of actions activity A_i in $Activities$
5: **if** (($s_i < a_i$) **and** (β can be executed with the resources of $Resources$)) **then**
6: $S' \leftarrow$ new state with the activities and actions of S plus the increment of 1 action of the activity A_i
7: $States.adding(S')$
8: `AddStates(`A_i`, `S'`, resources to run `β` in `$Resources$`, `$Activities$`)`
9: **end if**
10: **end for**

Using Algorithm 3, a new state S' is created by adding an action for the respective activity. Then S' is added to the $States$ and the method is called again in order to create new schedules based on S' and available resources. The following is an example that applies the scheduling algorithm to the partial order plan depicted in Figure 3.

4.1 Example of application of the scheduling algorithm

The representation of a state is described by a tuple (*complete, inProgress, notScheduled*). The component *complete* represents the activities already scheduled. Activities under scheduling are described in the component *inProgress*. The component *notScheduled* corresponds to the remaining activities. The activity of a plan is represented by its index followed by the number of actions it holds. These two information are separated by a hyphen. Initially in the plan of Figure 3, activity 0 is added to the *complete* component because it is in fact the number of resources available in the beginning. Thus, this state would be represented as follows: (0, , 1-1 1-2 2-3 2-4 1-5 6). The value of its heuristic is expressed by the maximum value of the next actions that can be scheduled. In this way, the heuristic is given by Max(225, 735, 510, 1570) = 1570. It is interesting to note that duration of activity 2 is not considered in this evaluation because it has a predecessor that should be executed first. Activity 6 represents the goal and has no duration.

Of all the possible combinations of states with the resources available in this iteration, the one which has the lowest heuristic is chosen. In this case it is (0, 1-1 1-5, 1-2 2-3 2-4 6) that was chosen based on the duration of the activity that finishes first (value of its duration is 225) and heuristic whose value is 225 + Max(735, 510, 1345) = 1570. The end value of activity 5 is considered in the evaluation of the heuristic even staying in the state *inProgress* because it is considered the end of any action from activity 1, which remains part of the action time from activity 5 calculated as 1570-225 = 1345. The resources of state previously presented are: *(1 Townhall*, *1 Peasant*, *0 Barracks*, *1 Supply*, *0 Footman*, *200 Gold*, *0 Wood* and *1 Peasan Borrow)*. The *peasant borrow* is being used by activity 5 and is available only when this activity is over. Besides the *peasant borrow*, there is a peasant free among the resources available that matches the end of the activity 1. This and the other states as possible choices generated in the first iteration of the algorithm are shown in figure 4.

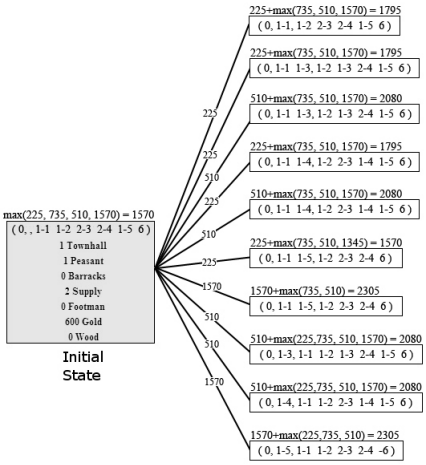


Fig. 4: First step from searching of SLA* to the plane of Figure 3.

This process continues, following the SLA* algorithm, until there are no more activities pending and the goal of resources is achieved.

5. Efficiency Strategies

To improve the efficiency of the SLA* for the scheduling task we developed some strategies to make it faster. In this section we describe three strategies: "Maximum of Actions", "Priority Use", and "Maximum of States".

5.1 Maximum of Actions

This strategy consists in creating states considering that one or more than one action could be scheduled if resources are available. This strategy can be implemented by making a small change in Algorithm 3. This change can be seen in Algorithm 4.

The change consists in introducing a variable $maximum$ into the code. If at a given moment no action could be scheduled, the value of maximum is equal to $true$. Otherwise, it is set to $false$. If $maximum$ is true, a predecessor state S is added to variable $States$. Using this strategy it is possible to

Algorithm 4 MaximumActions($A_j, S, Resources, Activities$)

1: $maximum \leftarrow true$
2: **for each** Active $A_i \in Activities$, such that $i \geq j$ **do**
3: $\beta \leftarrow$ action associated with activity A_i
4: $s_i \leftarrow$ number of actions activity A_i already scheduled in S
5: $a_i \leftarrow$ number of actions activity A_i in $Activities$
6: **if** (($s_i < a_i$) **and** (β can be executed with the resources of $Resources$)) **then**
7: $maximo \leftarrow false$
8: $S' \leftarrow$ new state with the activities and actions of S plus the increment of 1 action of activity A_i
9: `MaximumActions(`A_i`, `S'`, resources to run` β `in` $Resources$`, `$Activities$`)`
10: **end if**
11: **end for**
12: **if** $maximum = true$ **then**
13: $States.adding(S)$
14: **end if**

create only the states that can really contribute in the search for an optimal solution.

5.2 Priority Use

This mechanism, established when creating the partial order plan, consists in determining a priority for actions deemed most important to be carried out, i.e, actions that can reduce makespan. For example, an activity that has an action *build-peasant* should be executed before an activity *build-footman* because this helps in providing resources for other actions. As already mentioned, the priority does not define the order in which actions shall be scheduled. It only helps to improve the SLA* efficiency.

Besides improving the execution of the plan, the use of priorities contributes toward reducing the number of states that should be created. For the partial order plan in Figure 5, the first scheduling considers of activities 3, 4, 5, 7 and 9. The process may take long if all states are generated. It happens because some unnecessary schedulings can be generated. For example, it makes no sense to perform the action *collect-gold* of activity 9 while there are three actions of activity 3 available. It is more important to create a *peasant* than a *footman*. To eliminate some states apparently unnecessary a priority value is calculated for each activity. More details about how to obtain priority values for activities can be found in [2].

At each iteration when new states are created, the activities that have the same kinds of actions are grouped according to a priority value. As shown in Figure 5, for the scheduling of activities 3, 4, 5, 7 and 9, five states are created, each with one action of each activity. The strategy for this case is to consider the scheduling with two groups: *collect-gold* (formed by the activities 3, 5, 7, 9) and *collect-wood* (formed by activity 4). Therefore, only two states are created instead of five.

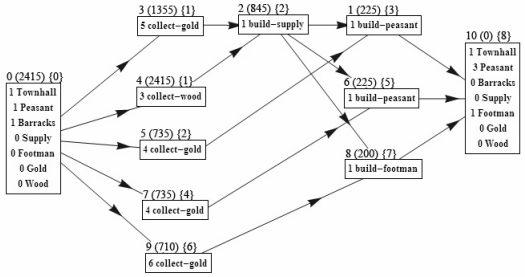


Fig. 5: Example of a partial order plan used in strategy "Use of Priority".

5.3 Maximum of States

This simple strategy basically sets up a maximum number n of states that can be created in each iteration of the scheduling. As the number of states to be considered is, in general, superior to n a cut should be taken. The cut should be made in such a way that the best promissor states be retained and the worst states be eliminated. To do this cut the initial value of the heuristic is considered; the lower the heuristic, the better is the state. In case of tie among the values of heuristics, the number of actions is considered. The states that contain more actions are considered the better ones. If after these analyses a tie persists then a random choice takes place. For example, if n = 3 for the scheduling from Figure 4, only the following states are created:

- State (0, 1-1 1-3, 1-2 1-3 2-4 1-5 6), with duration of 225 and heuristic equal to 1785.
- State (0, 1-1 1-4, 1-2 2-3 1-4 1-5 6), with duration of 225 and heuristic equal to 1785.
- State (0, 1-1 1-5, 1-2 2-3 2-4 6), with duration of 225 and heuristic equal to 1570.

6. Implementation Details

One of the difficulties in using the SLA* in scheduling is that some states during the search can be repeated. Two states are considered to be repeated if the successors are the same and this implies redundant updates. An obvious solution to avoid repeated states being created may require a search in the entire state space depending on the structure that stores them.

To solve the problem of repeated states during the search, we used a $Hashtable$ [7] that has as its primary Key an information describing a set of resources. The $Value$ associated to a key refers to another $hashtable$. This second $hashtable$ has as key actions that can be scheduled and action in progress. The value associated to a key describes a state. By using one $hashtable$ referencing another the search becomes more efficient.

Ten parameters are assigned for each state as follows:

- $completed$: Actions that have already been scheduled.
- $cost$: Duration / Cost of performing the scheduling.
- $countTotalAction$: Number of actions that are in progress (*inProgressCompleted* + *inProgressUncompleted*).
- $edge$: Members (schedulings) that are created considering the resources and actions available after this state. That is, they are the successor states.
- h: Heuristic value.
- $inProgressCompleted$: Actions that are in progress and will be finalized after the duration of this state (*cost*).
- $inProgressUncompleted$: Actions that are in progress but will not end after the duration of this state. In other words, part of the action is still in progress after the duration of this state (*cost*).
- $resourceStart$: Resources available at the beginning of this state.
- $resourceEnd$: Resources available at the end of this state. It takes into account the resources produced and released at the end of the actions *inProgressCompleted*.
- $toScheduling$: Actions that can be scheduled after this state.

Figure 6 shows the structure of $hashtable$s and states. $Hashtable$ 1 is formed by a resources key, which is a string that describes the quantity of each one of them. $Hashtable$ 2 consists of an actions key and this is a string which describes the quantity of them in every activity that can be scheduled, along with the actions in progress and not completed. The key is ordered increasingly according to the number of the activity. For example, the key "1 1 3 2 1 3 _1 1 200 1 4 150" has the following meaning:

1) Actions that can be scheduled:
   - 1 action of activity 1.
   - 3 actions of activity 2.
   - 1 action of activity 3.
2) Actions which are in progress but not completed:
   - 1 action of activity 1 with the duration of 200 remaining.
   - 1 action of activity 4 with the remaining of 150 duration.

With these structures it is possible to avoid the search algorithm making use of repeated states, which could decrease the efficiency of the process.

Another improvement was the use of a $heap$ structure to optimize the search for the best successor, which is the state that has the lowest heuristic. By using a priority queue based on a $heap$ it was possible to reduce the complexity of the search for the best successor from $O(n)$ to $O(1)$.

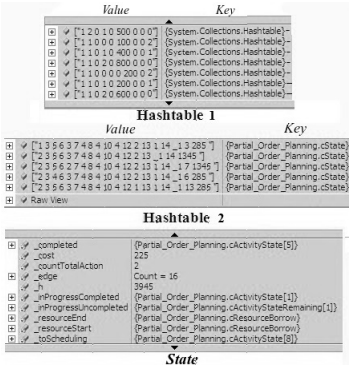


Fig. 6: Representation structure of the $hashtable$.

7. Experiments and Discussion of Results

The experiments were made on a Pentium Dual Core 2.66 GHz with 2 GB of Ram running Windows XP operating system. The CPU time spent on each problem was calculated by averaging 10 plays and it was measured in milliseconds (ms). For the tests we consider the following procedures: MEA, SLA* (1), SLA* (2), SLA* (3), SLA* (4).

MEA consists of a sequential plan and scheduling proposed in [6], [1]. SLA* (1) uses the partial order planning [2] with none of the improvements proposed in this work to the scheduling of actions. SLA* (2), SLA* (3), and SLA* (4) correspond to SLA* (1) equipped with strategies "Maximum of Actions" and "Use of Priority". In addition to these two strategies, SLA* (3) and SLA* (4) use the strategies "Maximum of States - 2" and "Maximum of States - 4", respectively.

Table 1: Results Experiment 1

	MEA	**SLA***(1)	**SLA***(2)	**SLA***(3)	**SLA***(4)
1	0,208	4221,578	11,912	11,207	11,846
2	0,211	4317,136	11,942	8,373	9,198
3	0,217	4302,851	9,342	8,575	9,450
4	0,210	4163,233	9,798	8,680	9,522
5	0,204	4419,502	13,757	8,740	17,342
6	0,204	4301,392	9,029	8,802	9,489
7	0,218	4228,612	17,760	11,580	9,486
8	0,208	4394,255	9,326	8,428	9,320
9	0,208	4363,905	9,003	12,093	9,549
10	0,208	4181,062	9,769	8,425	10,098
time (ms)	0,204	4289,353	11,164	9,490	10,530
makespan	16265	12920	12920	12920	12920

Table 1 shows the result of 10 runs based on the initial state *(1 townhall, 1 peasant)* and goal *(1 townhall, 5 peasant)*. As can be seen, the worst makespan is obtained by MEA, but its execution time is the smallest among the methods. Amongst the SLA* methods, SLA* (1) was the one that took more time to find the best solution. SLA* (2) required more time than SLA* (3) e SLA* (4) to find the best solution. SLA* (3) with a "Maximum of States" equal

two was the most efficient. All procedures with SLA* found the optimum solution.

Figure 7 shows the graphs for an experiment based on the initial state *(1 townhall, 1 peasant)*. The goal is to increase the number of *peasants*, from 1 to 20. Despite the time spent, the procedures based on SLA* managed to reach the best solutions in almost all executions.

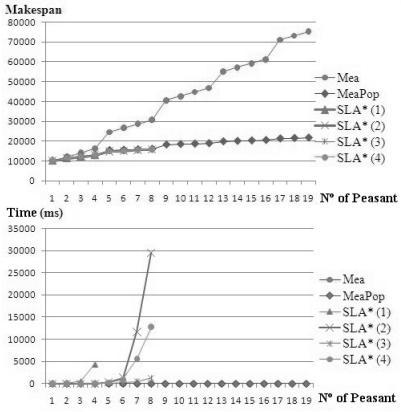


Fig. 7: Result of the experiments.

Again, MEA showed the lowest values of execution time in this test. However, as the problem became more complex, makespan values generated by MEA increased rapidly. With respect to makespan values, SLA* methods achieved better results than MEA.

In the next experiment we consider the initial state *(1 townhall, 1 peasant, 1 barracks)* and the goal is to increase the number of *peasants* to 5. The first goal is *(1 townhall, 1 barracks, 4 footman, 2 peasants)*, the second is *(1 townhall, 1 barracks, 4 footman, 3 peasants)* and so on until we reach 5 *peasants*.

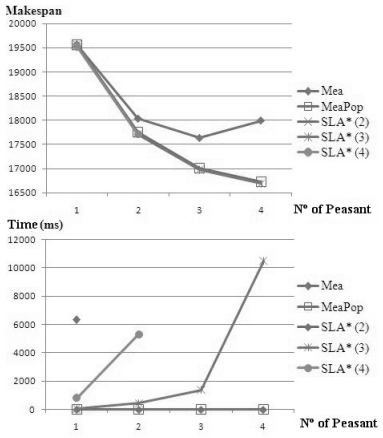


Fig. 8: Result of the experiments.

In the graph of Figure 8, SLA* (1) does not appear because it exceeded the maximum time in the first iteration. We established 30 sec as the maximum time for execution. Among the procedures based on SLA*, emphasis is given to SLA* (3) which generated expressive results with satisfactory execution time. It shows that efficiency strategies, such as "Maximum of States", should be used to increase the efficiency of the algorithm. The MEA has maintained its superiority in relation to time, but keeps makespan values far above the rest.

8. Conclusion and future work

In this paper we propose the use of a search algorithm and learning for the task of scheduling resources in RTS games. This type of research is recent and there are few studies devoted to it. The study was divided into two parts: the use of partial order planner developed in the work of [2] to create the action plan, and use of SLA* with improvements to schedule actions in the game.

Analyzing the results, despite the time spent by procedures based on SLA* being larger than the scheduler described in [6], [1], the makespan values obtained were better. Also it is important to note that this study allowed the use of search and learning in real-time, an important feature for domains including games and other dynamic environments. Our approach allows for a continuous improvement of the solution. Its employment makes it possible to obtain a solution within a time interval, which endows the system with reactivity.

As a future work, we aim to create a tool that assists the player in making decisions involving pooling of resources and attacks against the enemy. Also, other methods for search and approaches such SLA*T and genetic algorithms are going to be investigated.

Acknowledgments

This research is supported in part by the Coordination for the Improvement of Higher Education Personnel (CAPES), Research Foundation of the State of Minas Gerais (FAPEMIG) and Faculty of Computing (FACOM) from Federal University of Uberlândia (UFU).

References

[1] H. Chan, A. Fern, S. Ray, N. Wilson, and C. Ventura, "Online planning for resource production in real-time strategy games," in *ICAPS*, M. S. Boddy, M. Fox, and S. Thiébaux, Eds. AAAI, 2007, pp. 65–72.

[2] A. Branquinho and C. R. Lopes, "Planning for resource production in real-time strategy games based on partial order planning, search and learning," *SMC*, 2010.

[3] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[4] R. Zamani and L.-Y. Shue, "Solving project scheduling problems with a heuristic learning algorithm," *Journal of the Operational Research Society*, vol. 49, pp. 709–716, 1998.

[5] M. Buro and T. Furtak, "Rts games as test-bed for real-time research," *Workshop on Game AI, JCIS, 481-484*, 2003.

[6] H. Chan, A. Fern, S. Ray, C. Ventura, and N. Wilson, "Extending online planning for resource production in real-time strategy games with search," *Workshop on Planning in Games, ICAPS 2007*, 2007.

[7] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. The MIT Press, 2001.

Multifractal Phenomena in EcoSim, a large scale Individual-Based Ecosystem Simulation

Abbas Golestani
Department of Computer Science
University of Windsor
Windsor, Canada

Robin Gras
Department of Computer Science
University of Windsor
Windsor, Canada

Abstract— **This paper presents a multifractal analysis of EcoSim, a large evolving ecosystem simulation. Multifractal analysis of ecosystem time series using the Rényi fractal dimension spectrum demonstrates a self-similarity characteristic for a complex ecosystem. Results suggest the applicability of Rényi dimensions spectra to individuals' spatial distribution characterization for modeling empirical data and generating synthetic data by means of algorithms capable of generating such kind of measures. They also prove the capacity of our simulation to generate data with complex characteristics generally observed in real ecosystem studies.**

Keywords: Ecosystem simulation; Multifractal analysis; Rényi dimensions; Random walk model.

I. INTRODUCTION

Recently, researchers have begun to recognize the ecosystem signals as a highly nonlinear system [1]. Analysis of a time series with a high complexity, such as ecosystem time series which are result of interaction between individuals' behaviors, requires a nonlinear dynamical approach [2-5]. These methods are applicable to signals with a low dimensional deterministic nature. Most of the scientists believe that the chaotic behavior can be observed in many natural systems, such as the weather [6], so natural phenomena have to be considered as a chaotic system [7]. Therefore nonlinear dynamic methods based on the concept of chaos have been used to analyze ecosystem signals [8]. Dynamic studies of nonlinear systems allow us to describe the specification of biological processes [9]. Since the seminal work of Mandelbrot [10], many patterns and processes have proven to be efficiently described by fractals in many fields of the natural sciences. Fractal geometry and their resulting scaling properties have also been suggested as a way to characterize space-time heterogeneity in ecology [11]. Fractals identify the presence of patterns at multiple scales. Part of the appeal of fractals is that a single statistic can be used to describe potentially complex patterns in natural environments. The use of fractal geometry can be viewed as a tool to be used by landscape ecologists to aid in answering questions relating to scale [12]. Studies [13] have shown that natural phenomena present self-similar property over time steps.

A multifractal system is a generalization of a fractal system in which a single exponent (the fractal dimension) is not enough to describe its dynamics; instead, a continuous spectrum of exponents is needed. Self-similarity is a typical property of fractals. Scale invariance is an exact form of self-similarity where at any magnification, there is a smaller piece of the object that is similar to the whole [2]. Applications of multifractals to ecology still remain extremely anecdotic, limited to forest ecology [14-16], population dynamics [17], the characterization of species-area relationship, species diversity, and species abundance distribution [18-22], and the characterization of nutrient, phyto- and zooplankton patchiness [23-26]. Multifractal analysis techniques allow exploring features of signal distribution that are not considered very often [13]. Quantifying the complexity of the signal requires estimation of the fractal dimension spectrum where the complexity means higher variability in the general fractal dimension spectrum. Moreover, while the absolute value of a specific fractal dimension is highly affected by practical restrictions such as the limited number of data points, the variability of the fractal dimensions through the spectrum is less likely to be affected by these practical limitations. In this paper, the generalized Rényi fractal dimension spectrum is calculated and its variations have been studied because this method is the most accurate and common method for multifractal analysis [27, 28]. In most of the natural phenomenon the chaotic property and multifractal property exist together [29], [15].

We have conceived EcoSim, a large scale evolving predator-prey ecosystem platform that perform many study in theoretical biology and ecology. It has been shown in [8] that our simulation behavior is chaotic and in this paper we want to show that multifractal property also exists in our simulation. We analyze the time series corresponding to the variation of the number of individuals and individuals' positions. If we can detect and measure the underlying properties of these kinds of signals, it will be possible to build a model fitting the signal that can be used for prediction. Because this simulation is a logical description of how a simple ecosystem performs, this analysis can help biologists to better understanding of long-term behavior of ecosystem. We also apply our analysis to a random walk version of the simulation and compare it with the standard simulation. With this comparison it can be shown that multifractal characteristic happened only in this kind of complex simulation. The paper is organized as follows: in Section 2, we present our ecosystem simulation and random walk simulation. In Section 3, we explain the multifractal analysis. Finally, in Section 4, we present the results of applying multifractal analysis method to the two ecosystem simulation data.

II. THE SIMULATION

A. An Individual-based Evolving Predator-Prey Ecosystem Simulation Using Fuzzy Cognitive Maps as a Behavior Model

In this section, the main parts of the already existing evolving agent-based predator/prey ecosystem EcoSim are briefly introduced. The comprehensive description of this simulation has been proposed by [30]. This simulation is a logical description of how a simple ecosystem performs. In this simulation, complex adaptive agents (or, simply, individuals), each one of them using a Fuzzy Cognitive Map (FCM) as a behavioral model, are either a prey or a predator and a virtual world is implemented as a *1000* × *1000* matrix of cells.

1) Fuzzy Cognitive Maps

FCMs are a weight graph aiming to represent the causal relationship between concepts and to analyze inference patterns. In our simulation, the FCM is not only the base for describing and computing the agent behaviors, but also the platform for modeling the evolutionary mechanism and the speciation events. Each individual performs an action during a time step based on its perception of the environment. The FCM, called a map in our system, is used to model the agent behaviors (structure of the graph) and to compute the next action of the agent (dynamics of the map). A map contains three kinds of concepts: sensitive, internal, and motor. The activation level of a sensitive concept is computed by a fuzzification of the information coming from the environment (see Fig. 1). The activation level of the motor concept is used to determine what the next action of the agent will be, and a defuzzification of its value can be used to determine the amplitude of the action. Finally, the internal concept's activation levels correspond to the levels of intensity of the internal states of the agent and affect the computation of the dynamic of the map.

2) Intelligent Agents

Each agent has one FCM and several properties that determine its physical capabilities and its behaviors. The behaviors are determined by the interaction between the FCM and the environment. Each agent possesses its own FCM (coded by its genome, which is subject of the evolutionary process). The FCM contains sensitive concepts like foeClose, foodClose, energyLow, internal concepts like fear, hunger, curiosity, satisfaction, and motor concepts like evasion, socialization, exploration, breeding. It also contains links and weights representing the mutual influences between these concepts. The FCM of an agent is transmitted to its offspring after being combined with the one of the other parent and after the possible addition of some mutations. The behavior model of each agent is therefore, unique.

A very simple map can be defined to model an agent perceiving and reacting to its distance from a foe. The closer the foe, the more frightened the agent. Depending on this distance and also on the fear level, the agent will decide whether or not it will evade. The more frightened the agent, the faster the evasion. An FCM corresponding to this example is given in Fig. 1. In this example, there are two sensitive concepts: *foeClose* and *foeFar*, one internal: *fear* and one motor: *evasion*. There are also three influence edges: closeness to a foe excites fear, distance to a foe inhibits fear and fear causes evasion. Activations of the concepts *foeClose* and *foeFar* are computed by fuzzyfication of the real value of the distance to the foe, and the defuzzyfication of the activation of *evasion* tells us about the speed of the evasion. In our simulation each individual posses its proper map which contains around 30 concepts and hundreds of edges.

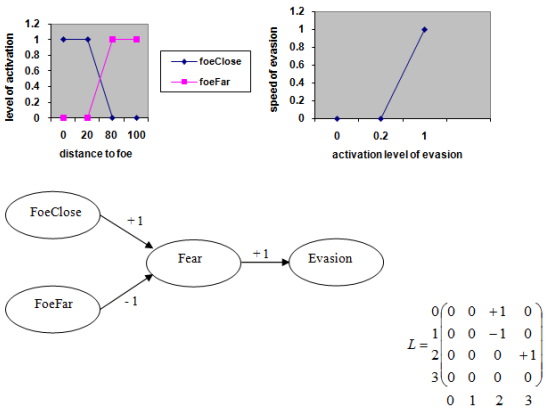


Figure 1. A simple fuzzy cognitive map for detection of foe and decision to evade with its corresponding matrix with 0 for "Foe close", 1 for "Foe far", 2 for "Fear" and 3 for "Evasion" and the fuzzyfication and defuzzyfication functions.

3) Species

In this simulation, a species is a set of individuals associated with the average of the genetic characteristics of its members. The average map of a species is computed based on the FCM matrices of all individuals' members of this species. It is considered that a species split if the difference between the maps of the two most dissimilar agents in the species is greater than a threshold; the threshold is the same for all species [30], [31]. Our speciation method consists in applying a 2-means clustering algorithm. With this process an initial species is split into two new species, each one of them containing the agents that are mutually the most similar.

4) Update

At each time step, the values of the states of all the parameters in the model are updated. The successive phases of the update process are as follows for each agent: perception of the environment, computation of all concepts of its map, application of their selected action and update of the energy level. Then, there is an update of the lists of agents, species and cells of the world. For each action which requires the agent movement, its speed is proportional to the level of activation of the corresponding action concept. Fig. 2 shows the population of prey and predator agents after each time step. Similarly Fig. 3 depicts the evolution of the number of prey and predator species in successive time steps. These patterns and the properties of the communities of species that are generated by our simulation have been shown to be very similar to the ones observed for real communities of species [32].

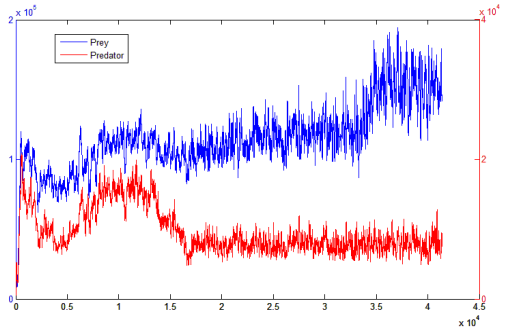


Figure 2. Population of prey and predator agents

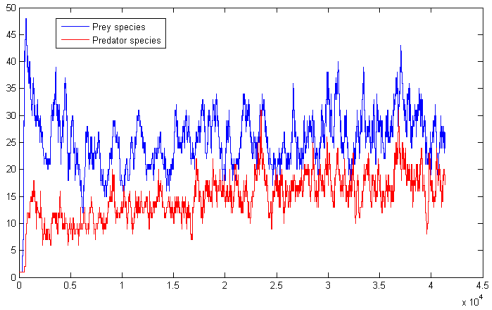


Figure 3. Evolution of the prey and predator species

A recent execution of the simulation produced approximately 40,000 time steps in 36 days. In the final generation, there are 190,000 preys, 22500 predators, 49 prey species and 19 predator species which are values close to the average ones observed all along the simulation process.

B. Random Walk Simulation

A random walk, sometimes denoted RW, is a mathematical formalization of a trajectory that consists of taking successive random steps [33]. The results of random walk analysis have been applied to computer science, physics, ecology, economics, psychology and a number of other fields as a fundamental model for random processes in time. In order to prove that the observed complex phenomenon measured by the generalized Rényi fractal dimension spectrum result from the interaction between the behavioral models of the individuals, we also develop a version of our simulation in which we do not use this behavioral model. Instead we apply a random walk process to our system. In the random walk version all the action such as eat, hunting, reproduction are replaced by a random movement action. The distribution of movements and the size of the world are the same as in the original simulation. In each time step, a specific number of predator and prey die and birth. In order to still have a realistic system, the Lotka-Volterra competition model with density dependence [34] has been used to compute these numbers. The following formula has been used to compute the variation in number of both of prey and predators.

$$\frac{dn_1}{dt} = r_1.\left(1-\frac{n_1}{k_1}\right).n_1 - a_1.n_1.n_2$$
$$\frac{dn_2}{dt} = r_2.n_2 + a_2.n_1.n_2 \qquad (1)$$

Where n_2 is the number of predator, n_1 is the number of prey, dn_1/dt and dn_2/dt represent the variation of the two populations against time, t represents the time; and r_1, a_1, r_2, a_2 and k_1 are parameters representing the interaction of the two species. We have analyzed the data that has been produced by our original simulation to determine the value of these parameters. With this mechanism our two models have therefore the same characteristic which is very important to make significant comparisons. The only difference between them is that with the random walk model, as there is no behavioral model that governs the agent's decision of action, we should not observe emerging spatial organization of population nor complex patterns of population variation.

III. Multifractal Analysis and Estimating Rényi Dimension Spectrum

The real fractal signals are not completely self-similar and thus may have different dimensions on different size scales and on different parts of the signals. When characterizing such a signal by a single dimension, we must decide on a range of scale lengths and on how to weight the contributions of its various scales to the global dimension. So we must specify a spectrum of dimensions for different values of scale lengths. Fractals that can only be fully characterized by specifying a spectrum of dimensions are called multifractals. A multifractal can be considered as an interwoven set of fractals of different dimensions, each having a different weight [35]. Thus it is more informative to define a multifractal spectrum of dimensions than to give a global fractal dimension. Such spectrum has proved to be useful in the analysis of heartbeat dynamics [36] and tropical rainforests [15].

For a simple fractal dimension computation, suppose we cover the attractor with squares of linear size r. Let Pi be the probability that a point is in square i. If there are N points on the attractor with N_i of them in square i, then $P_i = N_i/N$ and the capacity dimension is

$$D_0 = \lim_{r\to 0} -\frac{\log \sum_i P_i^0}{\log r} \qquad (2)$$

The sum $\sum_i P_i^0$ is then just the number of filled squares (there is no difference between squares with a different number of points because the power of P_i is 0). Thus D_0 weights all regions of the attractor equally; as power of P_i gets larger, the dimension computation gives emphasis to those regions where the measure P_i is large. We can define a generalized dimension:

$$D_0 = \lim_{r \to 0} \frac{1}{q-1} . \frac{\log \sum_i P_i^0}{\log r} \qquad (3)$$

The limit $q \to \infty$ gives the local dimension in the most densely populated region of attractor. For the special case of nonfractal or monofractals, there is no difference between the other dimensions in the spectrum. For multifractal systems, however, these dimensions are different, and the complexity of the system is determined by the shape and the variability of the fractal dimension spectrum. The Rényi dimension spectrum, as discussed in [11], gives the general formula for calculation of any fractal dimension D_q where $-\infty < q < \infty$. Rényi dimensions, is also called generalized dimensions [37]. Parameter q acts as a scanning tool scrutinizing the denser and rarer regions of the measure p_j [38-39]. For $q > 1$, regions with a high degree of concentration are amplified, while regions with a small degree of concentration are magnified for $q\_< -1$.

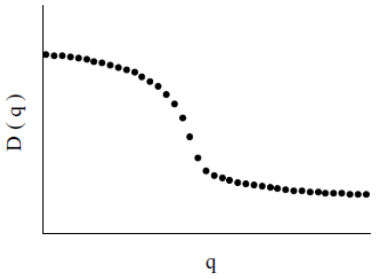


Figure 4. A typical Rényi dimension spectrum for a multifractal signal

For a signal to be characterized as multifractal, the Rényi dimensions spectra should be a nonincreasing function with a characteristic of sigmoidal form in general (see Fig. 4) but the most important feature is the nonincreasing pattern [38]. When studied distributions are close to monofractal measures (a uniform distribution of mass on a fractal set), Rényi dimensions spectra are closer to horizontal lines, so that $D_q \approx D_0$ [28].

IV. RESULTS

This section is an examination of simulation's population data and spatial distribution of individuals that are generated by two simulations: individual-based simulation and random walk simulation. Rényi dimensions spectra have been calculated for these signals. It will be shown whether these time series have a multifractal phenomenon or not. For each experiment, the data correspond to an average of five independent runs of the simulation. Each execution of the simulation used to generate the data for this analysis has produced approximately 16,000 time steps in 17 days by using the Sharcnet[1] resources. Table I displays the average and standard deviation of the number of prey and predator individuals during five runs of the simulation in the final time step.

[1] Shared Hierarchical Academic Research Computing Network

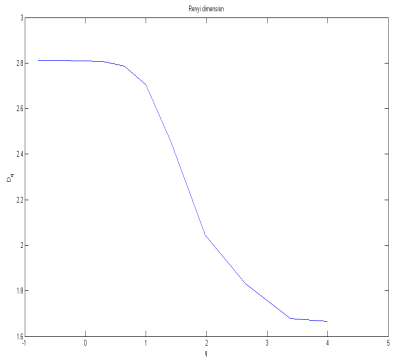
TABLE I. NUMBER OF INDIVIDUALS

	Average number of Individuals	Standard deviation of Individuals
Prey	**178000**	**9600**
Predator	**34000**	**2400**

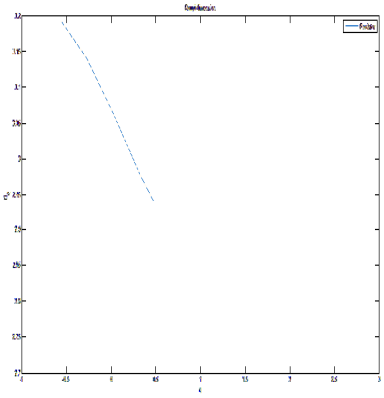
A. *Number of Individuals*

In this section we examined the multifractal property of the population size generated by both simulations: the behavioral model simulation and random walk simulation.

First, for the behavioral model simulation, the Rényi dimensions spectrum for both prey and predator population time series are generated and they are non-increasing (see Fig. 5). It can therefore be concluded that individuals' population time series generated by the behavioral model simulation have a multifractal structure. These results confirm previous results that come from analyzing real data. As said in the introduction, it has been shown that the multifractal phenomena exist in nature. The multifractal behavior also has been identified for real data such as population dynamics of soil microorganisms [40]. This result confirms that the population variation observed in our simulation contain complex multifractal patterns which is what is expected for of system to be able to model real ecosystems accurately.



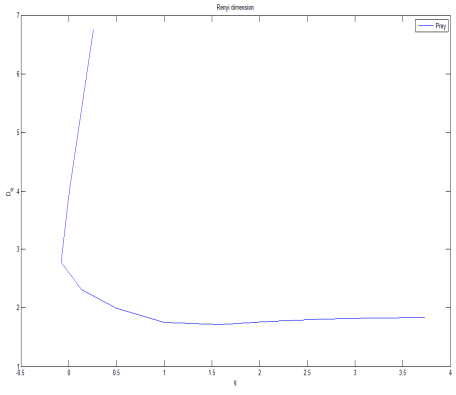
(a)



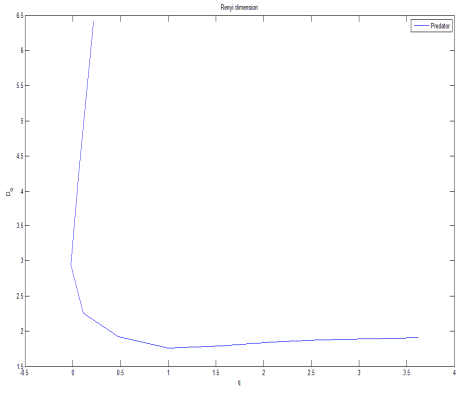
(b)

Figure 5. Rényi dimensions spectra curves for prey and predator population of original simulation

In contrast, for the random walk version of the simulation, the Rényi dimensions spectra for individuals' population distributions do not present a non-increasing shape. It means that there is no any multifractal characteristic in these signals (see in Fig. 6). Therefore, the assumption of complex self similarity for this kind of data is not possible. This result confirms that it is the interaction between the behavioral models of the agents that create the multifractal properties. A system very similar, with the same initial parameters and scales and with a non-linear model of population dynamic but without behavioral models is not enough to generate the complex pattern expected.



(a)



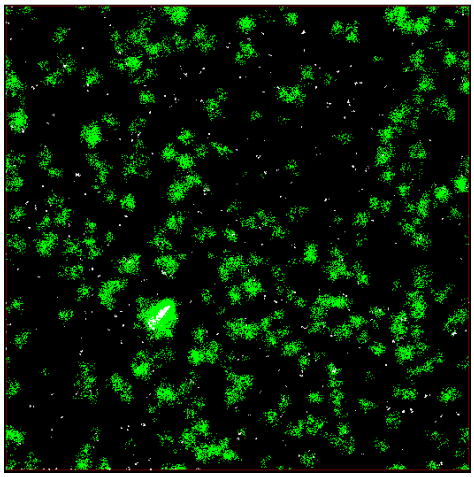
(b)

Figure 6. Rényi dimensions spectra curves for prey in Random Walk simulation

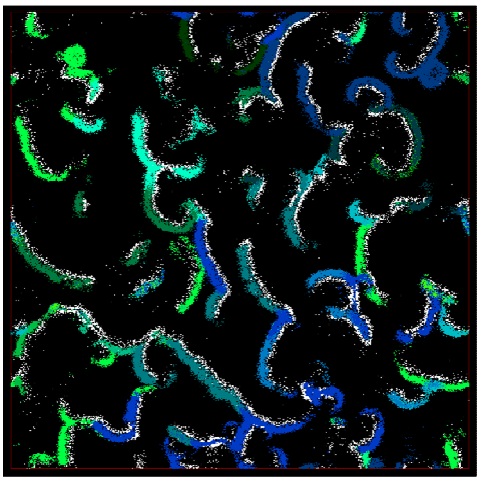
B. Spatial distribution of individuals

In this section we intend to investigate the characteristics of individuals' positions in our simulations. In other words, we apply multifractal analysis to the spatial distribution of individuals in the world (see section 2). The position of individuals for behavioral-model simulation and random-walk simulation has been shown in Fig. 7. Compared to the emerging herd patterns observed in the original simulation (7-b), the spatial distribution of individuals in the random walk version of the simulation seems somehow random (7-a).

The Rényi dimensions spectrum for spatial distribution of individuals for behavioral model simulation is non-increasing, so it can be concluded that the position of individuals also presents the multifractal structure (see Fig. 8-b).



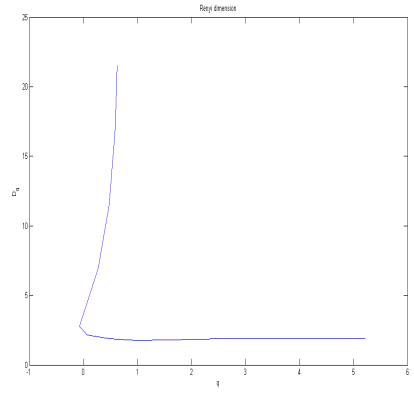
(a)



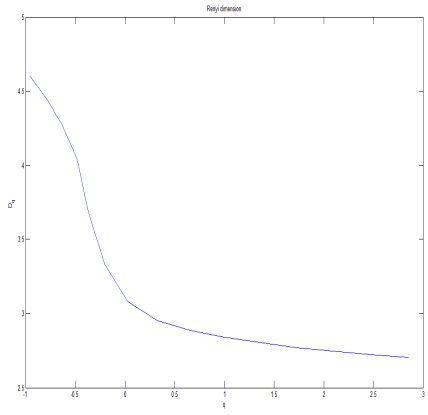
(b)

Figure 7. Spatial distribution of Individuals (a) Random Walk simulation (b) Complex simulation

The result of multifractal analysis also confirms that there is no such complex self-similar property between position of individuals in the random walk simulation (see Fig. 8-a).



(a)



(b)

Figure 8. Rényi dimensions spectra curves of spatial distribution for (a) Random Walk simulation (b) Complex simulation

Other real examples that have the multifratcal behavior are the swimming behavior of the calanoid copepod *Temora longicornis* [11], displacements of male Daphniopsis australis and microphytobenthos biomass distribution [13]. It has been shown that movement of these individuals can cause multifractal patterns which are close to what we have shown about our simulation individuals' movement effect.

Results indicate that individuals spatial distribution of our behavioral model simulation can be reasonable well fitted with multifractal models which is not possible for a random walk model with the same characteristics. The patterns observed also correspond closely to what can be found in natural systems. This result indicates that information achieved from Rényi dimensions might be used to model and simulate this kind of spatial distributions and the spatial distributions of agents in our simulation have the same complex emerging properties as real spatial distributions.

V. CONCLUSION

The purpose of this paper is the examination of the multifractal behavior of signals that are produced by EcoSim, a large scale ecosystem simulation. First, we gave a brief overview of the existing simulation. It is an individual-based evolving predator-prey ecosystem simulation, which uses fuzzy cognitive maps as a behavior model. Our purpose is to understand how complex and predictable our simulation is. The variations of the time series associated to ecosystem simulation are the result of complex simulation mechanisms. In this study, we focus on the time series corresponding to the variation of the number of prey and predator individuals and individuals' positions. We also applied our analysis to a random walk version of the simulation and compare it with the behavioral model simulation.

With this comparison it has been shown that multifractal characteristic happened only in behavioral model simulation. Individual spatial distribution could be fitted reasonably well with multifractal models. It showed that the data generated by our simulation present the same kind of multifractale properties as the ones observed in real ecosystems. It is also another important confirmation of the capacity of our simulation to model complex and realistic large scale systems. This fact also suggests the applicability of Rényi dimensions for the characterization of individuals' population distribution and for the simulation and modeling of empirical data by means of algorithms, such as iterated function systems, capable of generating such kind of measures.

There are a number of possible extensions of this study. We will be interested to analyze other versions of the current simulation or even other different simulations. We will be also interested to compare more into detail the correspondence between the patterns observed in our simulation with the ones coming from real data. This will give us the possibility to better understand what the important factors that lead to multifractal phenomenon are. Finally, we will be interested to see if

different kind of ecosystems can be characterized by different multifractal properties. We will be able to use our simulation to investigate different hypothesis and to see what properties of the system have influence on the multifractal patterns.

ACKNOWLEDGMENT

This work is supported by the NSERC grant ORGPIN 341854, the CRC grant 950-2- 3617 and the CFI grant 203617 and is made possible by the facilities of the Shared Hierarchical Academic Research Computing Network (SHARCNET: www.sharcnet.ca).

REFERENCES

[1] H. Kantz and T. Schreiber, Nonlinear Time Series Analysis. Cambridge University Press, 2003.

[2] C. J. Stam, "Nonlinear dynamical analysis of EEG and MEG: Review of an emerging field," Clinical Neurophysiology, vol. 116, pp. 2266-2301, 2005.

[3] M. E. Brandt, A. Ademoglu and W. S. Pritchard, "Nonlinear prediction and complexity of alpha EEG activity," International Journal of Bifurcation and Chaos, vol. 10, pp. 123-133, 2000.

[4] W. Kinsner, "A unified approach to fractal dimensions" International J. Cognitive Informatics and Natural Intelligence, vol. 1, no. 4, pp. 26-46, Oct-Dec. 2007.

[5] A. H. Meghdadi, W. Kinsner, R. Fazel-Rezai, "Characterization of healthy and epileptic brain EEG signals by monofractal and multifractal analysis". IEEE Canada 21st Canadian Conference on Electrical and Computer Engineering, 2008

[6] R. Sneyers, "Climate Chaotic Instability: Statistical Determination and Theoretical Background", Environmetrics, vol. 8, pp 517–532, 1997.

[7] L. Romanelli, M. A. Figliola, and F. A. Hirsch, "Deterministic Chaos and Natural Phenomena". Journal of Statistical Physics, vol. 53, Nos. 3/4, 1988.

[8] A. Golestani, R. Gras, "Regularity Analysis of an individual-based Ecosystem Simulation", journal of Chaos: An Interdisciplinary journal of Nonlinear Science, vol 20, 043120. pp 1-13, 2010.

[9] A. Golestani, A. Ashouri, K. Ahmadian, M. Jahed, M.A. Doostari. "Irregularity Analysis of Iris Patterns" WORLDCOMP'08 Congress (IPCV), pp 691-695, 2008.

[10] B.B. Mandelbrot, L. Pietronero, and E. Tosatti, "Self-affine fractal sets". Fractals in Physics, North Holland, Amsterdam, 1986.

[11] Milne. B.T, Lessons from applying fractal models to landscape patterns. In *Quantitative Methods in Landscape Ecology: The Analysis and Interpretation of Landscape Heterogeneity*, Turner, M.G. and Gardner, R.H., Eds. Springer-Verlag, New York, 199–235, 1991.

[12] B. T. Milne, Applications of fractal geometry in wildlife biology. In Wildlife and Landscape Ecology: effects of pattern and scale. Springer, pp 32-70, 1997

[13] L. Seuront, Fractals and multifractals in ecology and aquatic science. Ecology. CRC Press, 2009.

[14] I. Scheuring, and R.H. Riedi, "Application of multifractals to the analysis of vegetation pattern". *J. Vegetation Sci.*, 5, pp 489–496. 1994

[15] R.V. Solé, and S.C. Manrubia, "Self-similarity in rain forests: Evidence for a critical state". *Phys. Rev. E*, 51, pp 6250–6253, 1995.

[16] J.B. Drake, and J.F. Weishampel, Multifractal analysis of canopy height measures in a longleaf pine savanna. *Forest Ecol. Management*, 128, 121–127, 2000.

[17] J. Ozik, B.R. Hunt, and E. Ott, "Formation of multifractal population patterns from reproductive growth and local resettlement". *Phys. Rev. E*, 72, 1–15, 2005.

[18] C. Ricotta, "From theoretical ecology to statistical physics and back: Self-similar landscape metrics as a synthesis of ecological diversity and geometrical complexity". Ecol. Model., 125, 245–253, 2000.

[19] C. Ricotta, A. Pacini, and G. Avena, Parametric scaling from species to growth-form diversity: An interesting analogy with multifractal functions. BioSystems, 65, 179–186, 2002.

[20] L. Borda-de-Água, S.P. Hubbell, and M. McAllister, "Species-area curves, diversity indices, and species abundance distributions: A multifractal analysis". *Am. Nat.*, vol 159, pp 138–155, 2002.

[21] D.I. Iudin, and D.B. Gelashvily, Multifractality in ecological monitoring. *Nucl. Instr. Meth. Phys. Res. A*, 502, pp 799–801, 2003.

[22] [22] H. Laurie and E. Perrier, "Is species richness multifractal? Lessons from the Protea Atlas", *South African Journal of Botany*, 73, 297, 2007.

[23] M. Pascual, F.A. Ascitoti, and H. Caswell, Intermittency in the plankton: A multifractal analysis of zooplankton biomass variability. *J. Plankton Res.*, 17, pp 1209–1232, 1995.

[24] L. Seuront, F. Schmitt, Y. Lagadeuc, D. Schertzer, S. Lovejoy, and S. Frontier, Multifractal structure of phytoplankton biomass and temperature in the ocean. *Geophys. Res. Lett.*, 23, pp 3591–3594, 1996.

[25] L. Seuront and N. Spilmont, "Self-organized criticality in intertidal microphytobenthos patch patterns". *Physica A*, 313, pp 513–539, 2002.

[26] S. Lovejoy, W.J.S. Currie, Y. Teissier, M.R. Claereboudt, J.C. Bourget, and D. Schertzer, "Universal multifractals and ocean patchiness: Phytoplankton, physical fields and coastal heterogeneity". *J. Plankton Res.*, 23, pp 117–141, 2001.

[27] W. Kinsner, "Self similarity: The foundation for fractals and chaos", Technical Report, DEL95-2, University of Manitoba, 1995.

[28] E. Montero. "Rényi dimensions analysis of soil particle-size distributions". Ecol. Modell. 182, pp 305–315, 2005.

[29] A. Bershadskii. Some universal properties of multifractal chaos at nuclear giant resonance, Physical Review C, Volume 59, Number 6, 1999.

[30] R. Gras, D. Devaurs, A. Wozniak, and A. Aspinall, "An Individual-based Evolving Predator-Prey Ecosystem Simulation using Fuzzy Cognitive Map as Behavior Model", Journal of Artificial Life, vol 15(4), pp. 423-463, 2009.

[31] A. Aspinall and R. Gras. "K-Means Clustering as a Speciation Method within an Individual-Based Evolving Predator-Prey Ecosystem Simulation", Active Media Technology (AMT'10), 2010.

[32] D. Devaurs, R. Gras. "Species abundance patterns in an ecosystem simulation studied through Fisher"s logseries", *Simulation Modelling Practice and Theory*, Elsevier, pp 100 – 123, 2010.

[33] D. H. Barry, *Random walks and random environments*, Oxford University Press, 1996.

[34] N.S. Goel, S.C. Martia and E.W. Montroll, "On the Volterra and Other Non-Linear Models of Interacting Populations", *Academic Press Inc.*, 1971.

[35] J.C. Sprott, "Chaos and Time Series Analysis", Oxford University Press, 2003.

[36] L.A.N. Amaral, P.C. Ivanov, N. Aoytagi, I. Hidaka, S. Tomono, A.L. Goldberger, H.E. Stanley, and Y. Yamamato, "Behavioral-independent features of complex heartbeat dynamics". Phisical Review Letters 86, 6026-9, 2001.

[37] H.G.E. Hentschel, I. Procaccia, "The infinite number of generalised dimensions of fractals and strange attractors". Phys. D 8, pp 435–444, 1983.

[38] A. Chhabra, R.V. Jensen, Direct determination of the f (α) singularity spectrum. Phys. Rev. Lett. 62, pp 1327–1330, 1989.

[39] A.N. Kravchenko, C.W. Boast, D.G. Bullock, "Multifractal analysis of soil spatial variability". Agron J. 91, pp 1033–1041, 1999.

[40] J.W. Crawford, K. Ritz and I.M. Young, "Quantification of fungal morphology, gaseous transport and microbial dynamics in soil: An integrated framework utilizing fractal geometry". *Geoderma*, 56, 15–172, 1993.