SESSION

REAL-WORLD DATA MINING APPLICATIONS, CHALLENGES, AND PERSPECTIVES

Chair(s)

Dr. Mahmoud Abou-Nasr
Dr. Robert Stahlbock
Dr. Gary M. Weiss

Results of Mining Data Features During Computational Fluid Dynamics Simulations

Michael R. Gosnell†, Robert S. Woodley†, and Steven E. Gorrell‡
†21st Century Systems, Inc., 6825 Pine Street, Suite 141, Omaha, NE 68106, USA
‡Department of Mechanical Engineering, Brigham Young University, Provo, UT 84602, USA

Abstract— *Computational Fluid Dynamics (CFD) simulations provide a variety of data mining challenges and present an opportunity for novel solutions. A core challenge is that CFD computations can require weeks of computation on expensive high performance clusters, delaying investigation of results until a fully converged solution is obtained. 21st Century Systems, Inc. and Brigham Young University have been collaborating on a concurrent agent-enabled feature extraction project designed to mine feature data while a CFD simulation is executing. This paper summarizes the work and presents the combined results from a unified Industry/Application perspective. Empirical results show the concept validity and capability of obtaining CFD feature information much earlier than waiting for complete simulation convergence, which may ultimately save extensive computational resources and provide much quicker turnaround in development requiring CFD modeling.*

Keywords: Computational Fluid Dynamics (CFD), Subjective Logic, Feature Extraction, Decision Support, Concurrent Analysis

1. Introduction

Computational Fluid Dynamics (CFD) simulations numerically solve the governing equations of fluid motion to model and simulate a variety of systems and machines including ocean currents, atmospheric turbulence, combustion, aircraft, rotorcraft, and ship hydrodynamics. With increasing computational capabilities and the use of parallel codes, CFD simulations have increased in grid resolution and numerical accuracy to a point of correctly simulating highly complex fluid flow problems. Without appropriate data mining efforts in place, these large, time-accurate, three-dimensional computational models risk concealing rather than revealing the physics of interest.

Undertaking CFD efforts requires a few core steps: creation of a computer model, generation of a computational grid, computing a numerical solution, and data post-processing. Initial model creation and entry, along with construction of the computational grid, are pre-processing steps which are not typically suited for any data mining capabilities. A core challenge in CFD-related research is the computational resources required to obtain solutions, which can take days to months to reach full convergence. List [1] and Yao [2] have run unsteady Reynolds-averaged Navier-Stokes (URANS) simulations of gas turbine engine transonic fan stages with 166 million grid points and entire fans with over 300 million grid points respectively. The complementary challenge in CFD work is the amount of resulting data and associated time for analyses.

The obvious data mining application is with processing the terabytes of raw data following computation of the numerical solution. Many disciplines incorporating CFD research utilize software such as Evita, Intelligent Light's FieldView, and Kitware's ParaView to assist with post-processing and data mining of physical features within the CFD solution space. These types of programs are designed to post-process and visualize massive data sets and commonly include techniques such as feature extraction, construction of iso-surfaces, and automated visualization.

A more illusive opportunity for data mining was hypothesized to exist within the computational period of determining a CFD's numerical solution. The numerical solution achieved in the penultimate iteration is virtually identical to the final solution, so detection of flow features would be just as possible as detecting them within the final solution. Conceptually, detecting flow features in prior iterations of the solution would be attainable, but with increasing error. The risk is that before traditional convergence, features may not exist or conform to their accepted mathematical definitions. However, the tradeoff is that if certain features could be detected with appropriate levels of confidence, the CFD researcher might be able to obtain enough information to forego the continuation of the solution, saving CPU-hours. Figure 1 shows a conceptual view of the Concurrent Agent-Enabled Feature Extraction (CAFÉ) concept, trading off additional expense of concurrent feature detection with potential benefit of not requiring the CFD simulation to completely converge before items of interest are identified.

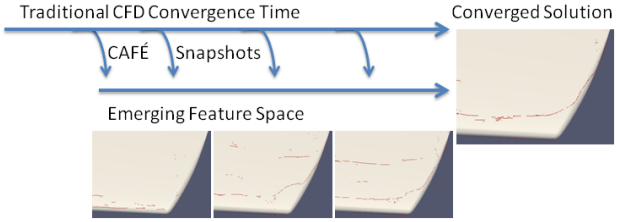


Fig. 1: CAFÉ concept showing concurrent feature mining

The CAFÉ concept was originally proposed as a partnership between 21st Century Systems, Inc. (21CSi) and Brigham Young University (BYU) and subsequently funded through Phase I and Phase II Small Business Technology Transfer (STTR) contracts. Following an initial investigation of feasibility, the concept was prototyped and expanded to include a variety of CFD features and data. The major findings of this work are summarized in this paper with an eye toward applications to external work.

2. CAFÉ Overview

The ultimate vision of CAFÉ spans CFD pre-processing, concurrent feature extraction and analysis, through to solution post-processing. The approach gains innovation as the solution was framed around an agent-based structure designed for decision support software applications. This structure allows all aspects of the CFD solution process to be included within the scope of CAFÉ, with the following high-level goals:

1) Provide concurrent feature extraction
2) Provide intelligent reasoning about extracted features
   a) Incorporate multiple extraction algorithms
   b) Determine the believability of features
3) Utilize detected features and results
   a) Hone search space to reduce resource waste
   b) Incorporate machine learning to generalize solutions and provide intelligent initial conditions

Goal 3 focuses primarily on CFD pre- and post-processing aspects and is beyond the realm of this discussion. Goal 1 focuses on one of the two main elements of the prototyping, concurrent feature extraction. Typical CFD simulations largely ignore the iterative solution data occurring before the required convergence. Concurrent aspects of CAFÉ utilize some of this intermediate data for analysis which is investigated while the CFD simulation continues toward a solution. Unlike final post-processing and analysis, CAFÉ must be able to make decisions on the feature extractions without the assistance of user input. This aspect is addressed in Goal 2 which utilizes results from multiple feature extraction algorithms, along with knowledge of the solution space, to provide intelligence about the detected features.

The two key CAFÉ architectural components for this presentation are the feature extraction and feature aggregation. Feature extraction takes the form of traditional CFD feature detection with additional analyses of uncertainty due to the feature space. Feature aggregation incorporates multiple extractions of the same identified feature to provide a single analysis of the feature space. In this manner, multiple algorithms can be incorporated with each being favored per its given strengths. CAFÉ performs extraction and aggregation using mathematically rigorous methods to determine when a feature is true or simply an artifact of an unconverged simulation.

2.1 Feature Extraction

Feature extraction algorithms form the core of the data mining approach by examining the raw data. In order to provide reasoning and decision support to the CFD users, CAFÉ uses an opinion space which captures characteristics of the system and allows for mathematical manipulation about multiple feature opinions. The tool employed to quantify the believability of a feature is encapsulated within subjective logic, developed by Jøsang [3]. This ternary logic captures belief (b), disbelief (d), and uncertainty (u) as an opinion, and intrinsically handles these in an algebraic space. These three elements are defined in [3] along with relative atomicity a to form an opinion or belief tuple ω (see Eq. 1). Operators within subjective logic allow consensus and discounting of opinions in such a way that combinations of opinions relating to features can be aggregated.

$$\omega_x = (b(x), d(x), u(x), a(x)) \mid b + d + u = 1 \qquad (1)$$

To form an opinion, each component of the belief tuple is given a numerical value, allowing the opinion to have an exact representation. To maintain uniformity and provide for mathematical constructs, the summation of an opinion's belief, disbelief, and uncertainty components is always equal to one with belief, disbelief, and uncertainty only taking on values between zero and one. Subjective logic is extremely attractive for incorporating the inherent uncertainty present during CFD execution as opinions are not forced to identify belief or disbelief. An agent can find, based on given information, how probable an outcome is rather than simply reducing the outcome to a binary TRUE or FALSE. In addition to the initial opinion formulation for feature detection, missing or incomplete data can also be incorporated within subjective logic by adjusting belief, disbelief, and uncertainty accordingly.

Each individual feature extraction algorithm in CAFÉ is tuned to generate an opinion based on the algorithm's strengths, weaknesses, and the relationship of the potential feature to the solution space. Specific details on each algorithm's opinions can be found in subsequent discussions and related work.

2.2 Feature Aggregation

As mentioned previously, the results of each feature extraction algorithm will be related to the solution space of a given situation. The focus of feature aggregation is to allow the strengths of individual feature detection, encapsulated through opinions, to provide intelligent feedback to the CFD user. Two subjective logic operators are fundamental in this aspect: the discounting operator and the consensus operator.

The discounting operator, defined by Jøsang [3], uses the symbol $\otimes$ written as $\omega_x^{AB} = \omega_B^A \otimes \omega_x^B$ where the superscripts represent the agent holding the opinion and the subscripts represent an agent, or piece of information, on which the opinion is based. In the above equation, a

discounted opinion of x is formed for A by A's opinion of B and B's opinion of x. Conceptually, the discounting of opinions allows individual, independent beliefs to be transferred along a chain of agents.

The counterpart to the discounting operator is the consensus operator. The consensus operator is used when multiple opinions are held about the same agent, or piece of information, and a single opinion is desired. The consensus operator, defined by Jøsang [4], uses the symbol $\oplus$ written as $\omega_x^{AB} = \omega_x^A \oplus \omega_x^B$ following the same syntax of the discounting operator. With supporting opinions, the consensus operator has the effect of reducing uncertainty.

CAFÉ's feature aggregation provides analysis through trust networks, built from opinions. A graphical representation of a trust network with two feature detection algorithms is shown in Fig. 2. The algorithm agent AA contains feature extraction algorithms with subscripts 1 and 2 denoting separate algorithms. The master agent MA combines information from multiple AAs to form its opinion on feature R.

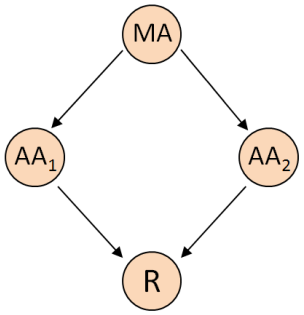


Fig. 2: Graphical representation of a two algorithm trust network.

Each AA forms its own opinion on R denoted by $\omega_{\mathrm{R}}^{\mathrm{AA}_1}$ and $\omega_{\mathrm{R}}^{\mathrm{AA}_2}$. The MA forms an opinion on each AA in use given by $\omega_{\mathrm{AA}_1}^{\mathrm{MA}}$ and $\omega_{\mathrm{AA}_2}^{\mathrm{MA}}$. Once the initial opinions are formed, they can be combined into a final opinion, $\omega_{\mathrm{R}}^{\mathrm{MA}}$, on the existence of a feature in R as

$$\omega_{\mathrm{R}}^{\mathrm{MA}} = \left(\omega_{\mathrm{AA}_1}^{\mathrm{MA}} \otimes \omega_{\mathrm{R}}^{\mathrm{AA}_1}\right) \oplus \left(\omega_{\mathrm{AA}_2}^{\mathrm{MA}} \otimes \omega_{\mathrm{R}}^{\mathrm{AA}_2}\right). \quad (2)$$

2.3 Decision Support Feedback

With an overarching goal of providing decision support to the CFD user, CAFÉ utilizes multiple feature extraction algorithms along with feature aggregation capabilities utilizing subjective logic opinions and the trust network framework. These key components allow for extracting features concurrent with CFD simulations and providing intelligent analysis of the feature space prior to convergence. While early feature extraction may contain large variations in the solution space, multiple sets of the solution space, taken many iterations apart, can also be incorporated within the trust network approach. Additional background on the CAFÉ architecture can be found in [5], [6] and a more thorough presentation of subjective logic is available in [4].

3. Vortex Core Extraction

Vortices are common occurrences in many types of engineering flows. They arise where there are large amounts of vorticity, or flow rotation. A vortex contains two interdependent parts: the vortex core line and the swirling fluid motion around the core. Many feature extraction algorithms have been developed to locate vortex core lines. Unfortunately, when extracting vortex core lines, there is not one markedly superior algorithm that correctly extracts all features within the spatiotemporal flow domain. Rather, there are multiple algorithms per feature that have been optimized for specific flow conditions. Roth [7] states, "none of the [vortex extraction] methods is clearly superior in all the tested data sets." This leaves a researcher with the significant problem of having to run a data set through multiple extraction algorithms and parse through the data output to find relevant features—which is exactly where CAFÉ's feature aggregation is paramount.

The initial CAFÉ work implemented two vortex core extraction algorithms. The first vortex extraction algorithm selected was the Sujudi-Haimes (SH) algorithm [8]. The SH algorithm was designed as a robust vortex core line detection algorithm for use in large 3D transient problems. It is commonly used in CFD post-processing software packages such as EnSight and pV3. The second vortex core extraction algorithm is the Roth-Peikert (RP) algorithm [7], [9]. The RP algorithm is specifically designed to extract fluid vortices in turbomachine simulations. What makes the RP algorithm unique and well suited for complex flow fields is the fact that it is designed to locate curved rather than straight vortex core lines. Thus, each algorithm is strongly suited to a different domain.

Table 1 gives the strengths, weaknesses and feature characteristics used for opinion generation using the SH vortex core extraction algorithm. The SH algorithm is specifically designed to extract straight vortex cores which is why a straight core factors into belief. Strength refers to the amount of flow rotation about the core and quality is a vortex characteristic defined by Roth in [7] (in this research, the angle between a vortex core line and its velocity vector).

Table 1: SH vortex core opinion generation components

Opinion Component	Contributing Factors
b	straight core, high strength, low quality
d	curved core, low strength, high quality
u	distance from possible trip point

The strengths, weaknesses and feature characteristics used for opinion generation using the RP vortex core extraction algorithm are given in Table 2. Setting a straight core as a weakness characteristic might be misleading because the RP algorithm does not extraneously extract straight vortex core lines. A straight core is incorporated as a weakness because when it comes to straight core lines there is more belief that

the SH algorithm will extract them correctly than the RP algorithm. Using the SH and the RP algorithms together in this fashion helps us to match each algorithms strengths with the flow situations for which they were designed.

Table 2: RP vortex core opinion generation components

Opinion Component	Contributing Factors
b	curved core, low strength, low quality
d	straight core, near zero strength, high quality
u	distance from possible trip point

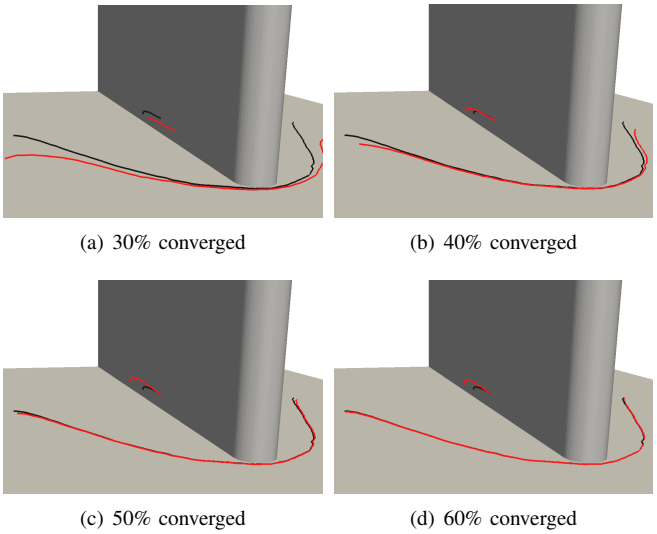
The feature characteristic used for the RP algorithm uncertainty is the same as the feature characteristic used for the SH algorithm which is distance from a possible vortex trip point. When using multiple feature extraction algorithms, the same feature characteristics are used for all algorithms since feature characteristics are not algorithm dependent.

A blunt fin geometry [10] was selected as one illustrative test case with clear, known vortex cores. Concurrent feature extraction was replicated by exporting and saving the entire flow field data set every 45 iterations from start to convergence at 900 iterations. Each of these saved data sets was input into the vortex core extraction method where vortex core lines were extracted using the RP and the SH algorithms—resulting in two feature extraction sets per saved data set. Agents then produced final opinions on all vortex features and a final aggregated feature set was produced.

One crucial piece of information needs to be clear for proper interpretation of results. When agents form opinions on extracted cores, they have information from the current iteration of the simulation and previous iterations only. They do not use information from the fully converged simulation, or any iterations beyond the current iteration, to form opinions on extracted cores. Belief, disbelief, uncertainty, and expected probability of vortex cores can be determined without requiring a final converged solution giving information about a final simulation's expected vortex cores before a simulation is 100% converged. However, Fig. 3 uses the final converged solution data to show the difference between concurrent extractions and the final solution.

Figure 3 compares concurrent vortex core extraction results obtained from the RP algorithm where the percent convergence is based on the number of iterations. The two blunt fin core lines are referred to as the "horseshoe" line, wrapping around the blunt fin, and the much shorter "fin" line. At 30% converged (Fig. 3(a)) the horseshoe line begins to take shape upstream. At 40% converged (Fig. 3(b)) the horseshoe line and the fin line are almost correctly resolved. At 50% converged (Fig. 3(c)) the end point of the fin line moves downstream. Already at 60% converged (Fig. 3(d)) the horseshoe line is spatially correct (but the fin line is not).

Figure 4 shows a graph of the feature displacement for the endpoints of the horseshoe core line and the fin core line extracted by the RP algorithm. The two vortex core lines exhibit similar behavior when they are extracted by the SH



(a) 30% converged (b) 40% converged

(c) 50% converged (d) 60% converged

Fig. 3: Comparison of RP extracted vortex core lines from the converged data set (black) and converging data sets (red)

algorithm. The start point is defined as the farthest upstream point and the end point is defined as the farthest downstream point. At 60% converged, all but the end point of the fin line has a non-negligible feature displacement. This shows that at 60% converged the entirety of the horseshoe vortex core line is very close to the same position it will be in at full solution convergence.

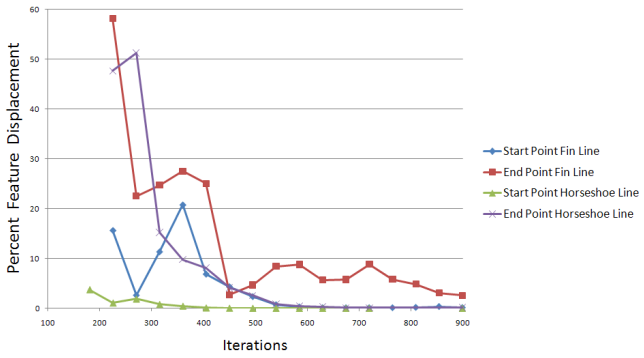


Fig. 4: Percent feature displacement for the endpoints of the horseshoe line and the fin line extracted by the RP algorithm.

As mentioned, behavior for SH core extraction was similar to RP results and resulting feature aggregation worked as expected. Additional investigation with vortex cores was performed including examination of more complex flow fields exhibited on a delta wing. This simulation was designed to match the experimental results of Kjelgaard [11] and the numerical results of Ekaterinaris [12]. The delta wing data revealed some distinct differences and advantage of extracting cores with multiple algorithms. Additional results and discussion of vortex core extraction are available in [6], [13].

4. Separation/Attachment Extraction

Separation and attachment lines are lines on the surface of physical bodies where the fluid flow abruptly moves away from, or returns to, the surface of a body. Two algorithms developed by Kenwright were included within CAFÉ prototyping: the Phase Plane (PP) algorithm [14] and the Parallel Vector (PV) algorithm [15]. The PP algorithm works by first finding the eigenvectors of the Jacobian matrix, then calculating and projecting critical points onto the phase plane. Depending on whether the local phase plane flow field is a saddle, repelling node, or an attracting node, a zero-crossing is determined and that point is marked as either a separation or an attachment line. The PV method compares the eigenvectors of the local velocity gradient tensor with the local velocity vector. Separation or attachment lines exist in this method when the local streamline curvature is zero.

The PP algorithm is better suited for unstructured grids, and it extracts disjointed line segments. The PV algorithm works well with curvilinear grids and extracts continuous line segments. Both algorithms are designed to detect straight lines and fail in detecting curved lines. Additionally, both methods work well when the extracted points reside in an area of high separation or attachment strength, commonly referred to as the pressure difference across the separation or attachment line. They also work best when the extracted points display a low velocity magnitude.

The selected algorithms appear to extract true separation and attachment lines accurately, but both suffer (to varying degrees) from extracting false lines. The original authors mention that this problem occurs "when flow separation/attachment is relatively weak and becomes diffused over several cells. This causes the phase plane algorithm to either detect multiple ghost lines or leave gaps." One cannot typically take these results alone as being completely accurate because of all the false extractions. However, with the incorporation of opinions to the feature extraction and aggregation, CAFÉ is able to provide feedback as to which lines are most likely to be correct.

One mathematical feature of working with subjective logic opinions is the ability to convert opinions into probability expectations. Being able to operate within the opinion space and convert results to a probabilistic space provides the opportunity for CFD users to quickly visualize the results with one common picture (as opposed to individually reasoning about the opinion components of belief, disbelief, and uncertainty). Figure 5 shows the probability expectation applied to both algorithms (showing the attachment lines only in this case) applied to a simulation of the swept ONERA M6 wing [16]. Both algorithms' probability expectation increases through the solution convergence with some of the more questionable areas (not true attachment lines) showing up with very low probability expectation. This method of analysis provides the researcher critical feedback on how the feature space is developing. The probability

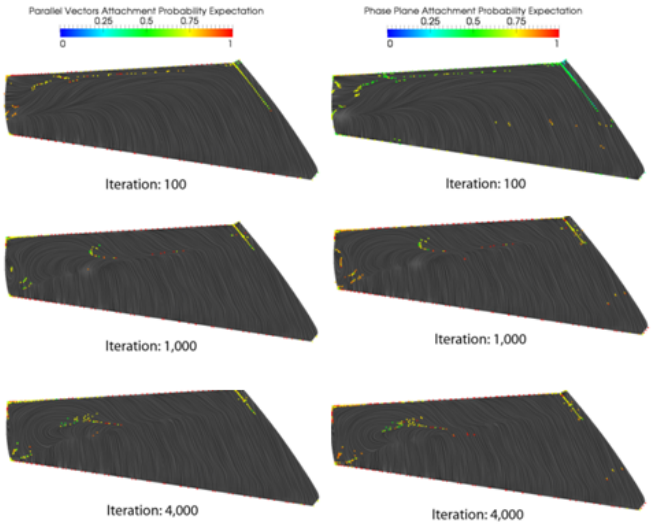


Fig. 5: Probability expectation of attachment lines from PV (left) and PP (right) algorithms at iterations 100, 1000, and 4000

expectation, along with experience, can be used within CAFÉ to determine the likelihood of features concurrent with the simulation. Additional information on separation and attachment line extraction is available in [17].

5. Shock Wave Extraction

Shock waves occur in fluid flow when the velocity of the fluid exceeds the speed of sound. Shock waves are characterized by sudden discontinuities in pressure, density, and velocity. Detection of a shock wave in CFD data is comparable to edge detection in image processing applications. Two shock wave extraction algorithms have been implemented in CAFÉ: the Lovely-Haimes algorithm [18] and the Ma-Rosendale-Vermeer (MRV) algorithm [19]. These two shock algorithms have outputs that are slightly different. The output of the Lovely-Haimes algorithm is a volume that encompasses a shock, while the output of the MRV algorithm is a surface designed to locate the shock exactly. Both of these algorithms are enhanced in CAFÉ through the use of multiple scalar values to compute derivatives, i.e. using both density and pressure instead of one or the other.

In addition to concurrent feature reasoning, CAFÉ shows additional strength in that the subjective-logic-based feature aggregation can be used on existing, processed data sets. This is illustrated by looking at an example converged CFD simulation of an ONERA M6 wing [16]. When the MRV algorithm is applied, false shock waves are detected as shown in Fig. 6. However, applying CAFÉ's intelligent feature extraction and applying opinions to the extraction based on algorithmic strengths and the simulation conditions allows the calculation of probability expectation. As with

the separation and attachment lines, probability expectation is correctly able to identify these false extractions as seen in Fig. 7. Once identified, thresholding could be applied to declutter the visualization, providing dynamic feedback to the researcher. This approach can leverage CAFÉ's capabilities onto previously executed simulations, providing better insight of detected features when traditional analyses might be misleading. Additional information on detecting shock waves is available in [17].

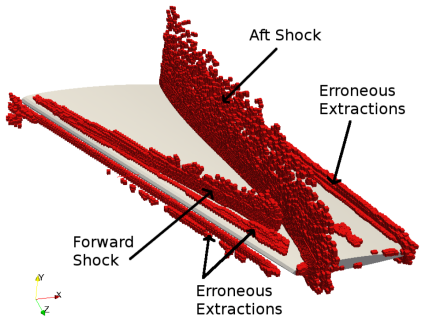


Fig. 6: Shock waves detected on a converted solution using MRV

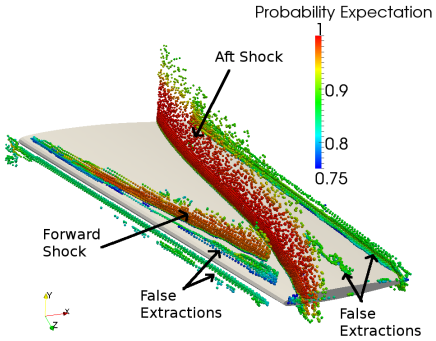


Fig. 7: Probability expectation of shock waves using MRV

6. Unsteady Vortex Core Extraction

Previous discussion of feature extraction has been entirely steady flows. In other words, calculation of the fluid flow was performed at a given snapshot in time. However, unsteady or transient flows—modeling the fluid flow over time—provides a much more accurate analysis of complicated systems such as turbomachinery.

Researchers have made modifications to the steady-state extraction algorithms in order to account for transient flow situations. Fuchs et. al. suggested the addition of time derivatives when extracting vortex core lines [20]. Lovely and Haimes derived a transient correction factor from the governing equations for their shock extraction algorithm which correctly extracts moving shock waves [21]. Weinkauf et. al. approached the extraction of moving vortices by using swirling particles and pathlines, which follow a particle in time instead of streamlines, in order to extract vortices in time-dependent simulations [22]. Each of these modifications has been shown to correctly extract features in unsteady data sets while the steady-state algorithms failed to reliably work in time-dependent simulations.

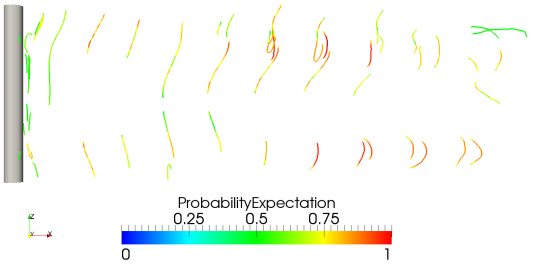
CAFÉ work included investigation of extending the methods to unsteady flows and implementation of unsteady vortex core extraction using the method presented in [20]. Unsteady vortex core extraction was implemented with RP and SH algorithms previously discussed. An additional aspect of feature tracking was required for analysis of features through time. The feature tracking method implemented during prototyping was the attribute-based method created by Reinders et. al. [23].

The strengths and weaknesses of Roth-Peikert and Sujudi-Haimes algorithms were clearly displayed in the cylinder data set. Extracted cores and probability expectation for both algorithms is shown for a single representative time slice in Fig. 8, showing the cores originating at the cylinder and moving downstream over time. Roth-Peikert performs well when extracting weaker cores, and as the vortex strength in most of the cores was quite low, it performed better, especially nearer to the cylinder. Some of the cores further downstream were also closer in agreement to the particles traced through time. However, both RP and SH failed to correctly extract cores as the cores broke up. An area where both algorithms fail is when the acceleration along a vortex core is not constant. As the cores were convected downstream, they were increasingly stretched, which caused a non-constant acceleration, so both RP and SH were shifted away from the vortex cell centers. Both algorithms also extracted cores that were less than half the height of the cylinder, a phenomenon which was observed by Zhang et. al. [24]. Additional information on extraction within unsteady flows is available in [25].

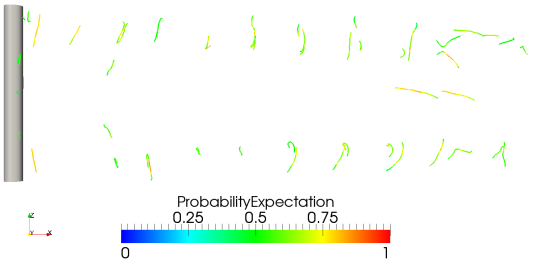
7. Conclusion

We have presented an overview of the CAFÉ concept along with validation of feature extraction capabilities both within steady state and unsteady CFD flows simulated in Fluent and OVERFLOW. Various aspects of CAFÉ were illustrated throughout the discussion. Within this work, BYU generated many datasets used to explore CAFÉ's capabilities for detection of steady and unsteady vortex cores, shock waves, and separation and attachment lines. Data sets generated for vortex core extraction included the blunt fin, delta wing, cylinder in cross flow, lid driven cavity, and NREL Phase VI wind turbine. Data sets generated for shock wave extraction were a supersonic ramp and swept ONERA M6 wing. Data sets generated for separation and attachment line extraction were cylinder in a cross flow and a delta wing.

We illustrated where CAFÉ could assist both with converged solutions as well as the original intent of concurrent



(a) Roth Peikert



(b) Sujudi Haimes

Fig. 8: Probability expectation of unsteady vortex core extractions

feature detection. Using CAFÉ to monitor the developing feature space during a lengthy simulation can alert the research to impending problem, saving time and resources. Applying multiple extraction algorithms with the capability of individual or aggregated visualization aids in understanding the detected feature space which can be applied to developing as well as converged solution sets. Our hope is that researchers incorporating CFD feature detection may be able to utilize concurrent feature detection and intelligent feature extraction and aggregation to operate more effectively.

Acknowledgement

This material is based upon work supported by the United States Air Force under Contract No. FA9550-10-C-0035 and is sponsored by the Air Force Research Laboratory (AFRL). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force.

References

[1] M. List, S. Gorrell, and M. Turner, "Investigation of loss generation in an embedded transonic fan stage at several gaps using high fidelity, time-accurate CFD," *ASME Journal of Turbomachinery*, vol. 132, no. 1, p. 011014, January 2010.

[2] J. Yao, A. Wadia, and S. Gorrell, "High-fidelity numerical analysis of per-rev-type inlet distortion transfer in multistage fans–Part II: Entire component simulation and investigation," *Journal of Turbomachinery*, vol. 132, no. 4, p. 041015, 2010.

[3] A. Jøsang, "A logic for uncertain probabilites," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 9, no. 3, pp. 279–311, June 2001.

[4] A. Josang, "The consensus operator for combining beliefs," *Artificial Intelligence Journal*, vol. 141, no. 1-2, pp. 157–170, October 2002.

[5] C. Mortensen, R. Woodley, and S. Gorrell, "Concurrent agent-enabled extraction of computational fluid dynamics (CFD) features in simulation," in *Proceedings of The 2009 International Conference on Data Mining*, July 2009, pp. 90–96.

[6] C. Mortensen, "A computational fluid dynamics feature extraction method using subjective logic," Master's Thesis, Brigham Young University, 2010.

[7] M. Roth, "Automatic extraction of vortex core lines and other line-type features for scientific visualization," PhD Dissertation, Swiss Federal Institute of Technology, 2000.

[8] D. Sujudi and R. Haimes, "Identification of swirling flow in 3-D vector fields," *AIAA 95-1715*, June 1995.

[9] M. Roth and R. Peikert, "A higher-order method for finding vortex core lines," in *Proceedings of IEEE Visualization*, October 1998, pp. 143–150.

[10] C. Hung and P. Buning, "Simulation of blunt-fin-induced shock-wave and turbulent boundary-layer interaction," *Journal of Fluid Mechanics*, vol. 154, pp. 163–185, 1985.

[11] S. Kjelgaard and W. Sellers, "Detailed flow-field measurements over a 75° swept delta wing," *NASA TP 2997*, 1990.

[12] J. Ekaterinaris and L. Schiff, "Vortical flows over delta wings and numerical prediction of vortex breakdown," *AIAA 90-0102*, 1990.

[13] C. H. Mortensen, S. E. Gorrell, R. S. Woodley, and M. R. Gosnell, "Data mining vortex cores concurrent with computational fluid dynamics simulations," in *Real World Data Mining Applications*, ser. Annals of Information Systems, M. Abou-Nasr, S. Lessmann, R. Stahlbock, and G. M. Weiss, Eds. Springer, Under Review.

[14] D. Kenwright, "Automatic detection of open and closed separation and attachment lines," in *Proceedings of the Conference on Visualization*, ser. VIS '98. Los Alamitos, CA, USA: IEEE Computer Society Press, 1998, pp. 151–158.

[15] D. Kenwright, C. Henze, and C. Levit, "Feature extraction of separation and attachment lines," *IEE TVCG*, vol. 5, no. 2, pp. 135–144, 1999.

[16] J. W. Slater, "ONERA M6 Wing," http://www.grc.nasa.gov/WWW/wind/valid/m6wing/m6wing.html, 2008.

[17] M. C. Lively, S. E. Gorrell, K. M. Hoopes, R. S. Woodley, and M. R. Gosnell, "Extraction of shock waves and separation and attachment lines from computational fluid dynamics simulations using subjective logic," *AIAA Paper 2012-1263*, January 2012.

[18] D. Lovely and R. Haimes, "Shock detection from computational fluid dynamics results," in *14th AIAA Computational Fluid Dynamics Conference*, vol. 1, 2000, pp. 296–304.

[19] K.-L. Ma, J. van Rosendale, and W. Vermeer, "3D shock wave visualization on unstructured grids," in *Proceedings of the 1996 Symposium on Volume Visualization*, San Francisco, 1996, pp. 87–94, 104.

[20] R. Fuchs, R. Peikert, H. Hauser, F. Sadlo, and P. Muigg, "Parallel vectors criteria for unsteady flow vortices," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 3, pp. 615–626, May/June 2008.

[21] D. Lovely and R. Haimes, "Shock detection from computational fluid dynamics results," AIAA Paper 99-3285, June 1999.

[22] T. Weinkauf, J. Sahner, H. Theisel, and H.-C. Hege, "Cores of swirling particle motion in unsteady flows," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1759–1766, November/December 2007.

[23] F. Reinders, F. H. Post, and H. J. Spoelder, "Visualization of time-dependent data with feature tracking and event detection," *The Visual Computer*, vol. 17, pp. 55–71, 2001.

[24] H.-Q. Zhang, U. Fey, B. R. Noack, M. König, and H. Eckelmann, "On the transition of the cylinder wake," *Physics of Fluids*, vol. 7, no. 4, pp. 779–794, April 1995.

[25] R. P. Shaw, S. E. Gorrell, R. S. Woodley, and M. R. Gosnell, "Vortex core line extraction and tracking from unsteady computational fluid dynamics simulations using subjective logic," *AIAA Paper 2012-1261*, January 2012.

Prediction of Pull-out capacity of Suction Caissons Using Self-Evolving Neural Networks

Abdussamad Ismail and Dong-Sheng Jeng

Abstract— **A self-evolving neural network is developed using a combination of PSO and JPSO algorithms to predict the pull-out capacity of suction caissons in clay. The algorithm is proposed with the aim of reducing the network complexity without compromising accuracy. A database consisting of experiments performed on suction caissons is used to construct and validate the network model. The performance comparisons indicate that the proposed self-evolving neural network predicts more the capacity of suction caissons accurately than neural networks developed using conventional methods.**

I. INTRODUCTION

SUCTION Suction caissons serve as cost effective alternatives to conventional offshore foundations such as driven piles. They are favorably used in deep ocean oil and gas developments due to construction difficulty associated with the installation of foundation in such environment. By virtue of their larger diameter, suction caissons give a better capacity to withstand lateral loads than piles. The construction of caissons involves allowing the caisson to sink into the sea bed under its own weight, and then subsequently undergo an assisted penetration through pumping out of water from inside the caisson. Suction caissons usually function as anchors to hold the offshore installations, subject to severe environmental conditions, in place. Thus, there is a tendency for pullout movement to occur due to tensile forces exerted by the chain attached to the caisson (see Figure 1). Accurate evaluation of the pull-out capacity of suction caissons is therefore necessary for a reliable geotechnical design of this kind of foundation. Various attempts to improve the understanding of the behaviour of suction anchors through physical and numerical modelling have been reported in the literature [1]-[2]-[2]-[4]-[5]. However, due to the limited information about the complex nature of failure mechanism involved the reliability of conventional methods of analysis in accurately predicting the capacity of suction anchors is challenged. In an attempt to improve the accuracy of pull out capacity estimation, Rahman *et al.* [6] developed an empirical model using BPN networks. Based on their finding, neural network models gave reasonably accurate results in comparison with observed capacities and FEM based predictions. The downside of developing models using such a conventional neural network design procedure is that there is tendency to end up with a sub-optimal network with undesirably large network size, which could undermine its ability to generalize.

In this the present work, an approach to simultaneous optimization of network topology and parameters is proposed. The aim is to minimize the size of the network while still maintaining accuracy and generalization capability. The proposed algorithm is used to develop an empirical model to predict the pull-out capacity of suction anchors. The model depends on parameters such as the caisson geometry (depth and width), un-drained shear strength of soil around the caisson tip, the depth of the load application point, direction of the pull-out force and loading rate.

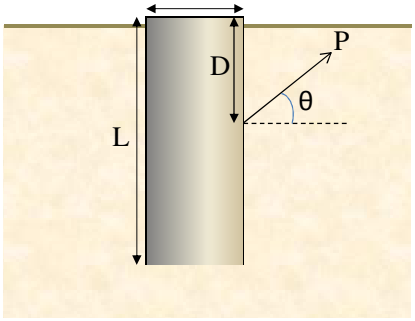


Figure 1: Suction caisson

.

II. NEURAL NETWORK MODELLING

Design of neural networks is a complex multi-dimensional optimization problem, involving not only choosing the optimum synaptic weights but also choosing a suitable processing function as well as an optimum network topology. The discrete, complex and multi-modal nature of the topology space, it is extremely challenging to optimize network architecture and the network parameters at the same time [7]. The classical topology optimization techniques include Network pruning [8][9], a top to bottom approach to network development, where the learning process begins with a large network, then subsequently trimmed to a smaller size by deleting redundant nodes and connections. Incremental learning algorithm [10][11] is a more convenient method in which the network size is increased by a gradual addition of nodes as training goes on. Near zero values are initially assigned to the synaptic weights associated with the newly added node to minimize the loss of knowledge. The

This work was supported by Nigerian Petroleum Technology Development Fund (PTDF).

Abdussamad Ismail is a research student with the Division of Civil Engineering, University of Dundee DD1 4HN, UK (phone: +447552872883 e-mail: azismail@dundee.ac.uk).

Dong-Sheng Jeng is a Professor in the Division of Civil Engineering, University of Dundee DD1 4HN, UK

drawback of both pruning and incremental learning algorithms is their tendency to get the network entrapped in the topology space local minima [12].

Later developments in topology optimization methodology are mainly associated with evolutionary concepts of combinatorial optimization. These include the genetic algorithm (GA) based topology optimization algorithms such as EPNet [13] and NEAT [14]. In EPNet, the population of networks is initialized by randomly generating the topology and synaptic weights of the networks. The networks are then subjected to a series of parametric and structural mutation steps. Parametric mutations take the form of partial training using back-propagation and simulated annealing, while the structural mutations involve addition of node or connection (growing) or removal of node of connection (pruning). The mutations cycles are repeated until a satisfactory network is obtained. While in ESPNet, inefficient cross over operation is avoided, the manner in which the topology population is developed makes it inferior, in terms of computational efficiency, to NEAT, which start with a population of smallest possible networks, then gradually increasing the complexity as training progresses. The weak point of NEAT lies in the intricate cross-over procedure involved while updating the network topology. The use of a combined PSO and DPSO algorithms in evolving neural networks have also been reported in the literature [15][16]. Although PSO based algorithms are simpler to use and computationally more efficient than GA based methods, the inefficient procedure of topology generation undermines the capability of the algorithms to arrive at the optimum solution.

In this research, a population based self evolving network is proposed, where the initial topology begins with a single hidden node, then gradually evolving in size as the training progresses. The self evolution process begins by generating a population of neural nets with each having a random set of connection and synaptic parameters. The connection parameters are binary, assuming a value of 1 if there is a connection between two nodes and 0 if otherwise (Figure 2). They are updated using a jumping particle swarm optimization (JPSO) procedure in the cause of optimization process. JPSO algorithm, developed by Martınez-Garcıa and Moreno-Pe´rez [17] is discrete optimization technique that turned out to be more efficient than the discrete version of particle swarm optimization algorithm (DPSO) proposed by kennedy and Eberhart [18]. In Jumping PSO (JPSO) algorithm, the particle jumps from its current position to a new position under the influence of particle's experience, global best position or explorative tendency which make a particle to make a random explorative search (see Figure. 3). Whether a particle jumping is influenced by previous experience or by explorative tendency depends on chance. The particle's position is updated as follows:

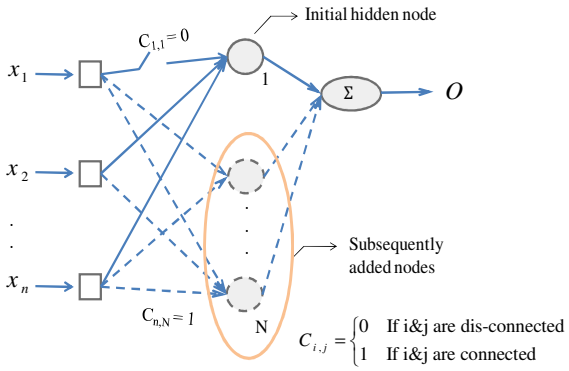


Figure 2: topology of self-evolving network

$$\boldsymbol{x}_{t+1} = c_1 \otimes \boldsymbol{x}_t \oplus c_2 \otimes \boldsymbol{b} \ \oplus c_3 \otimes \boldsymbol{g} \qquad (1)$$

where $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+1}$ are the vectors of current and future particle positions in the discrete search space. The parameters c_1, c_2 and c_3 are probabilities of jumping randomly, towards the best particle position and to the best swarm position respectively. B and g are, respectively, the particle best and global best positions. Equation (1) is implemented as follows:

$$x_{i,t+1} = \begin{cases} x_{i,t} * \rho & P_{x_{i,j}\to\rho} = c_1 \\ x_{i,t} * b_i & P_{x_{i,j}\to b_i} = c_2 \\ x_{i,t} * g_i & P_{x_{i,j}\to g_i} = c_3 \end{cases} \qquad (2)$$

$$c_1 + c_2 + c_3 = 1$$

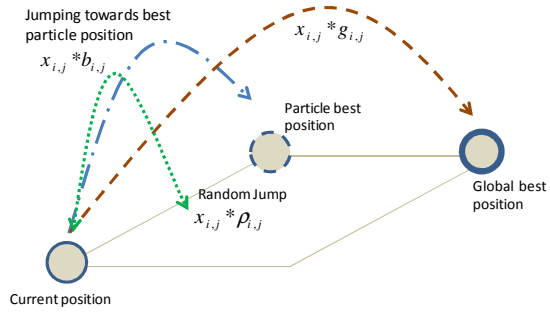


Figure 3: Graphical representation of jumping particle in topology space

ρ represents a random value. The * operator is implemented by a stochastic modification of the features of the current particle with some features of its attractor. The updated position determined using equation (2) could be worse than the current one, therefore a random local search is carried out to find a better solution. In this research due to the mixed nature of optimization problem involving both continuous and discrete variables, the local search is carried out using few steps of back-propagation algorithm. Also, due to random resetting of the position of a fraction of swarm population at intervals during the training, the c_1 is reduced to zero, and the values of c_2 and c_3 sum up to 1. The

proposed JPSO algorithm is represented by the flowchart in Figure 4.

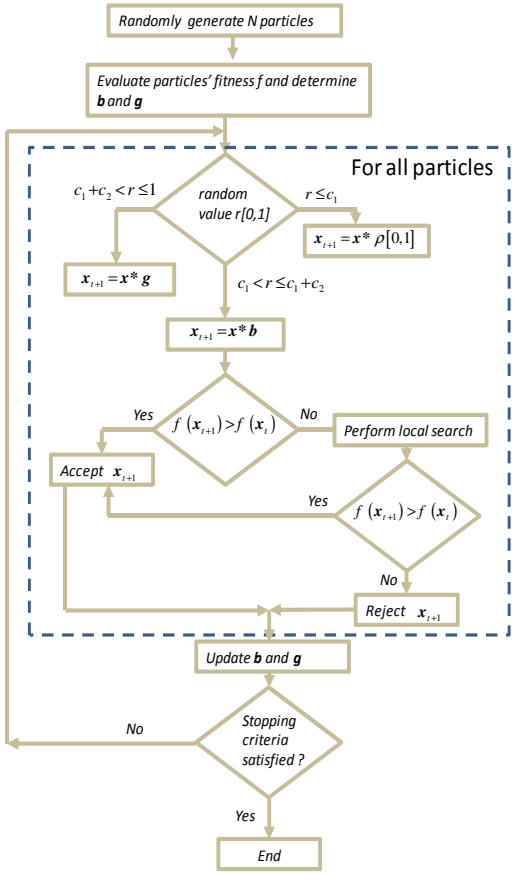


Figure 4: Flowchart describing JPSO alorithm

The synaptic weights of individual networks in the population are updated using a combination of PSO and BP algorithm. For the PSO part, the network parameters are updated using the following equations as proposed by Clerc and Kennedy [19]:

$$\boldsymbol{v}_{i,t+1} = \chi[\boldsymbol{v}_{i,t} + c_1 r_1(\boldsymbol{b}_i - \boldsymbol{x}_{i,t}) + c_2 r_2(g - \boldsymbol{x}_{i,t})]. \quad (3)$$

$$\boldsymbol{x}_{i,t+1} = \boldsymbol{x}_{i,t} + \boldsymbol{v}_{i,t+1} \quad (4)$$

χ is defined as:

$$\chi = \frac{2}{\left|2-\varphi-\sqrt{\varphi^2-4\varphi}\right|} \quad (5)$$

where $\varphi = c_1 + c_2 > 4$. The advantage of putting together the two techniques is to take the advantage of global search capability of the former and the ability of the later to perform local search. The proposed hybrid optimization algorithm is based on PSO technique and BP algorithm, whereby, both algorithms are used successively as training progresses. The idea is to get the best out of the two powerful algorithms by developing such algorithm which integrates more efficiently the PSO and BP techniques. The algorithm involves initially training the network parameters using PSO for a certain number of iterations, then training some selected (best performing) particles among the swarm population using BP algorithm for few number of iterations. The results of the local search by BP algorithm are then used to update the positions of relevant particles and the PSO takes over again. The cycle is repeated until a sufficiently accurate is obtained. In PSO, there is a possibility of particles to cluster around one co-ordinate, thereby, causing a stagnation in the search progress. To avoid this problem, duplicate particles have their positions reset randomly at the end of each cycle of PSO iterations. The positions of least performing fraction of the swarm population are also randomly reset in order to improve the topology search, having removed the random jumping aspect of JPSO.

When no further improvement is observed, the complexity of the network is increased by adding more nodes, one node at a time. To prevent the destruction of the so far acquired knowledge, the previous best particle positions (both topology and synaptic weights) are retained while adding one more node to the members of the swarm population. In this way, having to deal with unnecessary large network size is avoided as the case is with the models proposed by Kiranyaz et al. [15] and Xian-Lun et al. [16], while at the same time avoiding the danger of getting stuck in the local minima of topology space as in the case of classical pruning and incremental learning techniques. The algorithm of self-evolving network is summarised in the following steps:

1. Initialize a population of N neural networks with a single hidden node and randomly generated set of synaptic weights and connection parameters. Regard the population as particle swarm of N size with each network as a particle.
2. Select the best particles and update their positions for few iterations using BP algorithm.
3. Evaluate the fitness of each particle and update the best particle and global positions..
4. Use PSO to update the weight vector of each particle
5. Use JPSO to update the binary connection parameters of each particle.
6. Update the particle best position and the best swarm position
7. Use PSO/JPSO to update particle co-ordinates for certain number of iterations in the following sub-steps:
   a. Use PSO to update the weight vector of each particle
   b. Use JPSO to update the binary connection parameters of each particle.
   c. Update the particle best position and the best swarm position
8. If convergence is sufficient then stop. Else continue
9. Reset randomly the binary and continuous parameters of duplicate particles. Also, reset in the same manner, the binary parameters of certain fraction of the swarm with poor fitness.
10. Select best particles and update their continuous parameters using some steps of BP. If the training is satisfactory go to step 11. Else continue.
11. If number of iterations < maximum then go back to step3. Else continue

12. Generate N particles with one additional node over the current number of nodes. Replace all current particles with newly generated particles while retaining the current particle best positions (topology and synaptic weight). Then go back to step 3.
13. Terminate algorithm and return result.

Furthermore, a parallel swarm population with of fully connected networks but with the same number of nodes is optimized alongside the swarm with partial connections. The purpose of the fully connected swarm is to assist the partially connected swarm in the search for best network. The partially connected swarm can therefore learn from the fully connected swarm whenever the best swarm position in later is more accurate than the former.

Activation function

The choice of suitable activation function is pivotal to a successful development of neural networks. Sigmoid function has been the most widely used model for ANN development due to its stability. However, despite its popularity, it is not the optimum for all circumstances [19][20]. In this research, a combination of linear and product unit functions are used as processing functions. the idea behind selecting the two functions is to come up with a relatively simple and tractable for the relationship between input and output parameters at the end of the network training the processing function used is expressed in the following equation as:

$$f(\boldsymbol{x}) = k_1 c_1 \left(\boldsymbol{w}_1^T \boldsymbol{x}\right) + k_2 c_2 \prod_1^n x_i^{w_{2i}} \quad (6)$$

where n is the number of inputs; c_i is an adaptive coefficient, while k_i is a binary coefficient; $\boldsymbol{x}$ is the vector of inputs to the node; $\boldsymbol{w}$ represents the vector of synaptic weights of input signals. The binary coefficient assumes the value of 0 when function is switched off and a value of 1 when the function is turned on. In the training process the binary coefficient is updated together with connection parameters, while the adaptive parameter is updated alongside the synaptic weights using PSO-BP hybrid algorithm. In this manner, the topology, the synaptic weights and the activation functions are simultaneously optimized.

III. NETWORK DEVELOPMENT

A. Suction caisson data

The data used to in this research consist of 62 pull out test data sets compiled by Rahman *et al.* [6] from various sources in the literature. The data consists of caissons of various dimensions embedded into clayey soils, subject to pull-out forces in vertical, horizontal and inclined directions. Table 1 contains the database summary.

TABLE 1
STATISTICAL PROPERTIES OF SUCTION CAISSON DATABASE

Parameter		Training	Testing
L/d	Average value	1.56	1.642
	Standard deviation	0.7818	0.7675
	Range	0.23 - 4	0.23 – 4
s_u	Average value	12.1	11.22
	Standard deviation	10.3187	9.697
	Range	1.8 - 38	1.8 – 38
T_k	Average value	0.0024	0.0018
	Standard deviation	0.0092	0.008
	Range	1E-05 -0.04	1E-05 -0.04
Θ	Average value	67.7	79.64
	Standard deviation	37.594	28.692
	Range	0 – 90	0 - 90
D/L	Average value	0.0781	0.0292
	Standard deviation	0.1883	0.136
	Range	0 - 0.69	0 - 0.68
q_u	Average value	87.67	93.608
	Standard deviation	79.9813	85.6374
	Range	12.9 - 387.2	12.9 - 370.4

B. Input parameters

The pull-out capacity of suction caisson in embedded in clay deposit depends on various parameters such as un-drained shear strength in the case of clays (s_u), depth of embedment (L), caisson diameter (d), the direction of pull-out force (θ), depth of load application from the top of the caisson (D) and the non dimensional loading rate parameter (T_k), which is a function of soil permeability and the velocity of pull-out. The aforementioned parameters are organised into a set of five parameters which control the ultimate capacity of suction anchor (q_u) as represented by equation (7).

$$q_u = f\left(\frac{L}{d}, \frac{D}{L}, T_k, s_u, \theta\right) \quad (7)$$

The parameters on the right hand side of equation 5 serve as inputs to the network, while the ultimate resistance to pull-out is the output of the network.

C. Networks Training and validation

The database was partitioned into training and testing sets. A total of 37 data sets were used for training, while the remaining 25 sets were reserved for testing. To enhance the ability of the network to generalize, the data is split in such a way that both the training and testing data, in a statistical

sense, belong to the same population. For the purpose of comparison, several network the conventional BPN network was also trained, alongside the proposed self-evolving network.. The training was brought to a termination when the quality of prediction cease to improve with further training effort with the view to avoiding over-fitting. The parameters used in assessing the prediction quality in the case of both training and testing are the root mean square error (RMSE) and the coefficient of determination (R^2).

The optimized network can be represented more simply by the following empirical relationship:

$$q_u\,(kPa) = 21.2801\left(\frac{L}{d}\right)^{-0.1018} s_u^{1.04791} T_k^{-0.0294} (1+\sin\theta)^{0.3357}\left(1+\frac{D}{L}\right)^{0.3048} - 3.6474\left(\frac{L}{d}\right)^{1.0874}(1+\sin\theta)^{5}\left(1+\frac{D}{L}\right)^{0.357} + 150.517\left(\frac{L}{d}\right) - 31.891 s_u + 428.2415 T_k + 142.7912(1+\sin\theta) - 316.42 \quad (5)$$

(8)

Figures 5(a)-(b) display the prediction results of the optimized network plotted against the training and testing data respectively. It can be clearly seen from the figures that the network gives a good correlation with both training and testing data. Most data points seem to fall within 15% envelope. The performance of the optimized network is compared with sigmoid network (BPN), product-unit network (PUNN) and a fully connected network with a combination of linear and product unit processing functions in Table 2. The proposed model seems to outperform all the networks considered with the highest value of R^2 (0.9810) with respect to testing data despite being the smallest network. This shows the proposed algorithm is capable of knocking out redundant nodes and synaptic links that could undermine generalization. It is also noteworthy that BPN had to use almost twice the number of parameters used the self-evolving network but still does not achieve accuracy of the later.

TABLE 2(a)
CONFIGURATIONS OF VARIOUS NETWORKS CONSIDERED

Type of network	No of nodes	Number of network parameters
Self-evolving Net	2	15
BPN	4	28
PUNN	3	18
Lin+PUNN	2	18

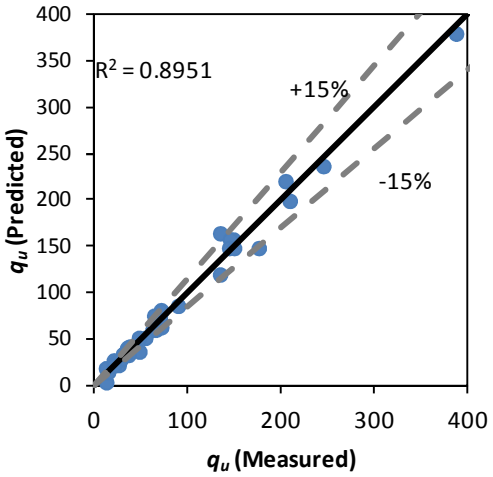


Figure 5(a): Comparison of measured pullout capacity (training data) and self-evolving model predictions. The ultimate capacity is in kPa

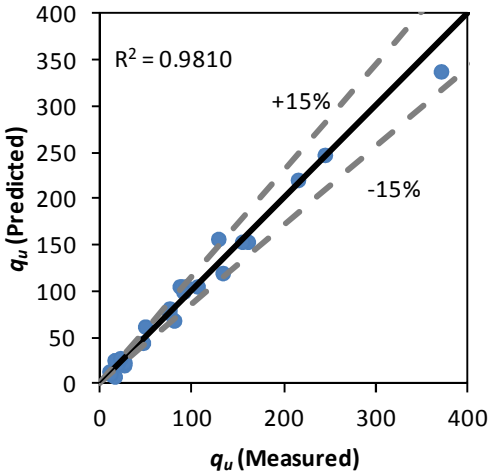


Figure 5(b): Comparison of measured pullout capacity (testing data) and self-evolving model predictions. The ultimate capacity is in kPa

TABLE 2(b):
SUMMARY OF TRAINING AND TESTING RESULTS FOR VARIOUS TYPES OF NETWORK

Type of network		Training	Testing
Self-evolving Net	N-RMSE	0.025704	0.04189
	R^2	0.9851	0.981
BPN	N-RMSE	0.01644	0.06838
	R^2	0.9939	0.9778
PUNN	N-RMSE	0.031641	0.047287
	R^2	0.9774	0.9731
PUNN+Lin	N-RMSE	0.029101	0.05835
	R^2	0.981	0.9634

To further assess the quality of predictions, the ratio of predicted capacity to measured capacity $\left(\lambda = q_{u\ Predicted} / q_{u\ Measured}\right)$ is used. The mean $\left(\mu_{\lambda}\right)$ and standard deviation $\left(\sigma_{\lambda}\right)$ of ratio λ give a great deal of insight about the reliability of model prediction. The mean value of λ ratio indicates whether a model, on average, underestimates or overestimates the value in question. The standard deviation gives a measure of scatter in the prediction. A perfect model with 100% accuracy will have a mean value of 1.0 and a standard deviation of zero. From the bar chart in Figure 6, It can be seen that the optimized model gives the best estimate of ultimate capacity on average. The model has a slightly higher value of scatter than PUNN model. However, on the overall the optimized model yields the best result as the PUNN over estimates the capacity by over 25% (against the optimized model with only 5% overestimation).

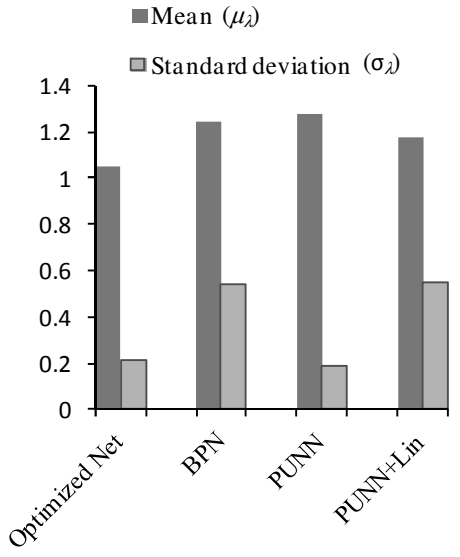


Figure 6: Comparison of the models based on the mean and standard deviation of λ ratio

To examine the influence of various parameters involved in the modeling of ultimate capacity, a sensitivity analysis is carried out by removing a parameter from the input set and evaluating the model performance without the parameter. The procedure is repeated until all input parameters considered are covered. The results are shown in Figure 7. It can be seen from the figure that the soil shear strength su is the most significant parameter affecting the pull-out capacity. The least significant variable is the loading factor T_k..

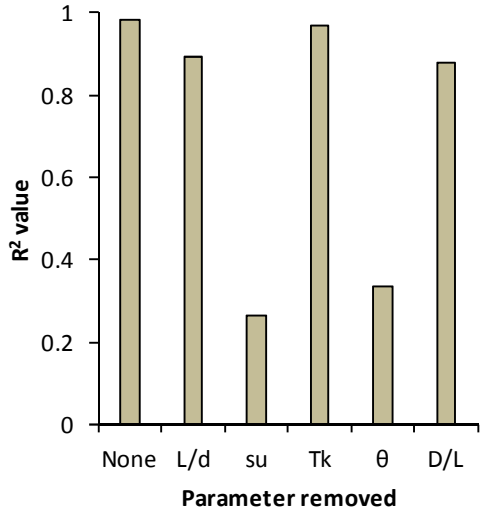


Figure7:Model sensitivity to various input parameters

IV. CONCLUTIONS

The simultaneous optimization of topology and synaptic weights of neural networks is a desirable but highly challenging task. While the traditional methods are inefficient, the bio-inspired population based algorithms are lacking in computational efficiency due to the random generated topology population with possibly many redundant connections and nodes. In this paper, a self-evolving network capable of growing from small to more complex network is developed. The key features of the algorithm are the ability to grow from a very small network to a complex without a loss of information while maintaining the capability of exploring the search space.

The proposed algorithm is implemented to predict the ultimate pull-out capacity of suction caisson penetrating into clay. The soil shear strength, the caisson's geometry, the loading conditions are used as inputs to the model. Based on the performance comparisons, the proposed model, with smaller network size, gives a more reliable estimate of ultimate capacity of suction anchors than BPN, PUNN and the combination of PUNN and linear models.

REFERENCES

[1] Finn W.D.L and Byrne, P.M. (1972). " The evaluation of the breakout force for a submerged ocean platform." Pro-ceedings, O€shore Technology Conference. OTC 1604: 351-65

[2] Fuglsang, L.D. and Steensen-Bach, J.O. (1991). " Breakout resistance of suction piles in clay." Proceedings of the international conference: centrifuge 91. Rotterdam, The Netherlands A. A. Balkema., 153-9

[3] Rao, S.N., Ravi R, and Prasad, B.S. (1997). " Pullout behaviour of suction anchors in soft marine clays." Marine Georesources and Geotechnology., 1(15):95-114.

[4] Zdravkovic, L., Potts, D.M., Jardine, R.J., 2001. A parametric study of the pull-out capacity of bucket foundations in soft clay. Geotechnique 51 (1), 55–67.

[5] Chairat, S, Randolph, M. and Gourvenec, S. (2004). "Inclined Pull-out Capacity of Suction Caissons." Proceedings of the Fourteenth International Offshore and Polar Engineering Conference.

[6] Rahman, M.S., Wang, J., Deng, W., Carter, J.P. (2001). "A neural network model for the uplift capacity of suction caissons." Comput. Geotech. 28: 269–287.

[7] Miller, G. F., Todd, P. M. and Hegde, S. U. (1989). "Designing neural networks using genetic algorithms." Proc. 3rd Int. Conf. Genetic Algorithms and Their Applications, J. D. Schaffer, Ed. San Mateo, CA: Morgan Kaufmann, pp. 379–384.

[8] Reed, R. (1993). "Pruning algorithms—A survey." IEEE Trans Neural Netw 4(5):740–747

[9] Chandrasekaran, H., Chen, H. H. and Manry, M. T. (2000). "Pruning of basis functions in nonlinear approximators." Neurocomput, 34:29–53

[10] Dunkin, N., Shawe-Taylor, J. and Koiran, P. A. (1997). "New incremental learning technique." Proceedings of the eighth Italian workshop on neural nets (Neural Nets Wirn Vietri-96.). Springer Verlag. 112–8

[11] Bahi, J. M., Contassot-Vivier, S. and Sauget, M. (2009). "An incremental learning algorithm for function approximation."Advances in Engineering Software 40: 725–730.

[12] Angeline, P. J., Saunders, G. M. and Pollack, J. B. (1994). " An evolutionary algorithm that constructs recurrent neural networks, IEEE Transactions on Neural Networks, 5, 54/65

[13] Yao, X. and Liu, Y. (1997). "A New Evolutionary System for Evolving Artificial Neural Networks." IEEE Transactions on Neural Networks, 8-3:694-713

[14] Stanley, K. and Miikkulainen, R. (2002). "Evolving Neural Networks through Augmenting Topologies." Evolutionary Computation, 10(2): 99-127

[15] Kiranyaz, S., Ince, T., Yildirim, A. and Gabbouj, M. (2009)."Evolutionary artificial neural networks by multi-dimensional particle swarm optimization." Neural Networks (2009) in press

[16] Xian-Lun T., Yon-Guo L. and Ling Z. (2007). "A hybrid particle swarm algorithm for the structure and parameter optimization of feedforward neural networks. " LNCS 4493:213-218.

[17] Mat`ınez Garc`ıa ,F. J. and Moreno P´erez J. A. (2008) "Jumping Frogs Optimization: a New Swarm Method for Discrete Optimization." , Technical Report DEIOC 3/2008, Department of Statistics, O.R. and Computing, University of La Laguna, Tenerife, Spain

[18] Kennedy, J. and Eberhart, R. C. (1997). "A Discrete Binary Version of the Particle Swarm Algorithm." Proceedings of IEEE Conference on Systems, Man, and Cybernetics, iscataway, New Jersey,USA. 4104–4109.

[19] Sopena, J.M., Romero, E. and Alquezar, R. (1999); "Neural networks with periodic and monotonic activation functions: a comparative study in classification problems." Ninth International Conference on Artificial Neural Networks (ICANN '99). 1:323 - 328.

[20] Wong, K.-W., Leung, C.S. and Chang, S.-J. (2002). "Use of periodic and monotonic activation functions in multilayer feedforward neural networks trained by extended Kalman filter algorithm." IEEE Proceedings. Image Signal Processing, 149 (4), 217 – 224.

Model for Aggregated Water Heater Load Using Dynamic Bayesian Networks

M. Vlachopoulou[1], G. Chin[1], J. C. Fuller[1], S. Lu[1], and K. Kalsi[1]
[1] Pacific Northwest National Laboratory, Richland, WA 99354 USA

Abstract - *The transition to the new generation power grid, or "smart grid", requires novel ways of using and analyzing data collected from the grid infrastructure. Fundamental functionalities like demand response (DR), that the smart grid needs, rely heavily on the ability of the energy providers and distributors to forecast the load behavior of appliances under different DR strategies. This paper presents a new model of aggregated water heater load, based on dynamic Bayesian networks (DBNs). The model has been validated against simulated data from an open source distribution simulation software (GridLAB-D). The results presented in this paper demonstrate that the DBN model accurately tracks the load profile curves of aggregated water heaters under different testing scenarios.*

Keywords: Dynamic Bayesian network, water heater, demand response, smart grid

1 Introduction

New advances in power and energy technologies have recently accentuated the need for revision of the current power grid operation to ensure reliability and performance. The next generation power grid, known as the "smart grid", provides a new framework that includes the new technology deployments and addresses the issues of system state uncertainty and deregulation [1]. Demand response (DR), distributed generation (DG) and distributed energy storage (DES) are basic strategies applied during the smart grid operation. They formulate a new power grid paradigm that incorporates distributed architecture instead of the traditional centralized one, as well as dynamic response to real time changes of the power grid state. The organizations and corporations involved in the power generation, transmission and distribution will be required to have the necessary analysis and planning tools for a successful transition to smart grid operation.

The demand response feature is enabled by allowing devices and appliances to modulate their operation in response to an event causing a change of state of the voltage and frequency of the grid, energy prices, or a number of other factors. The Federal Energy Regulatory Commission (FERC) in [2] specifies different types of DR programs including dynamic pricing without enabling technology, dynamic pricing with enabling technology, direct load control and interruptive tariffs. A simple overview of DR strategies can be found in [3]. Multiple utility companies have expressed interest in assessing the impact that DR can have to their operations. Results of DR studies using empirical data are presented in [4] and [5]. The dynamic behaviour of the smart grid stems from the new perception of the grid as a network with real-time communication of its components. The DR strategies support the required network response flexibility, however it is essential that their application preserves and enhances the grid reliability. Reliability studies under DR operation [6] and [7] have used the DC Optimal Power Flow (OPF) model to access the impact of DR programs. A general impact of DR, DES and the penetration of renewable energy resources to the smart grid reliability is analysed in [8].

There is a prominent need for analysis tools that can be used by utilities and other power grid management participants, to forecast the DR effect on the power grid operation. Load forecasting is an important component of this analysis, where load is the power sink of an appliance or device. There are different types of load forecasting, depending on the time horizon of the forecast. There have been certain critical factors determined that affect such forecasts [9]. The forecasting methodologies range from statistical methods, like regression and time series analysis, to artificial intelligence and data mining methods, like neural networks, fuzzy logic and support vector machines. Efforts have been recently made for analytical modelling of aggregated loads [10]-[12].

This paper presents a novel approach of forecasting aggregated end-use water heater load in residential areas. This approach entails a Dynamic Bayesian Network (DBN) for modelling of the aggregated load behaviour. The developed model successfully and accurately emulates the behaviour of the aggregated water heater load due to two factors. First, the DBN structure enables modelling of the dynamic physical system behaviour. Second, end-use load information data have been used for training of the network. Currently the DBN has been trained and tested using simulated data produced by simulation software (GridLAB-D). This DBN model can provide the basis for an accurate and flexible tool that is deployed for the analysis of DR strategies. It is easier and faster to use compared to

GridLAB-D and also provides a larger degree of flexibility since it can be retrained using different data sets.

This paper is organized as follows. In Section 2, an overview of the DBN principles of operation and application examples are given. Section 2 also includes a brief description of the software which was used to generate the training and testing data for the DBN analysed in this paper. In Section 3, a detailed description of the DBN model, as applied to the problem of water heater load aggregation, is given. The results illustrating the performance of the DBN model can be found in Section 4. Finally, Section 5 includes the conclusions of this research and future work.

2 Dynamic Bayesian network principles

Bayesian networks are a widely used machine learning methodology with diverse areas of application like medical diagnosis, sensor modeling and reliability analysis [13]. Studies of dynamic Bayesian networks are relatively more recent and aim in modeling a constantly changing system. In this section, a description of the basic structure and principles of operation of DBNs is presented. Additionally, the simulation software that provided the training and testing data is described.

2.1 Dynamic Bayesian networks

A Bayesian network (BN) or belief network is a probabilistic graphical model. In a BN, nodes represent random variables and directed arcs represent conditional dependencies. Every random variable has an associated conditional probability table which contains the probabilities of the variable being assigned to specific values or states based on the values of parent variables. These probabilities are commonly derived from collected data or prior knowledge. Once a BN has been constructed, the values of certain variables can be set based on evidence or observations. The posterior probabilities of the query variables can then be computed given the set of evidence variables as knowledge. Inferencing refers to the propagation of the evidence through the network followed by computation of the updated probabilities of the query variables.

For temporal analysis, a dynamic Bayesian network may be used to model the stochastic evolution of a set of variables over time. In a DBN, discrete time is introduced and conditional distributions are related to parent variable values of the previous time point. Since current events cause future events, but not vice-versa, directed arcs always flow forward in one direction in a DBN. For many applications, the graphical representation of a DBN often takes the form of a first-order Markov or hidden Markov model. DBNs have been used in a variety of applications in areas such as speech recognition [14], distributed sensor networks [15], and computational biology [16].

In developing a BN or DBN model, domain expertise is invaluable in a number of modeling steps. First and foremost, the structure of the BN in terms of the variables and conditional dependencies rely heavily on expert input. The structure of a BN should resemble the logical or physical topology of the system or process that it is modeling. BN structure learning algorithms including score-and-search-based and constraint-based methods are also available to automatically generate BN structures from training data, but it has been found that such algorithms are most effective in verifying a manually-constructed BN rather than constructing a BN from scratch. Expert input is also important in defining variable states, as they should represent the specific conditions of logical or physical entities in the BN. With respect to the conditional probability tables, we have found that BN parameter learning algorithms such as maximum likelihood estimation and expectation-maximization are mostly effective in learning probabilities. After parameter learning, however, we typically have experts verify that the learned probabilities appear reasonable.

2.2 GridLAB-D simulation

The DBN training and testing data were produced by the simulation software, GridLAB-D. GridLAB-D is an agent-based, open-source, power grid simulation tool developed at Pacific Northwest National Laboratory (PNNL) for the Department of Energy (DOE) to simulate the complexities of the smart grid from the substation to the end-use load [17].This allows users to develop models to simulate the behaviors of individual end-use loads and their interactions with the power system, including voltage effects, weather dependencies, control functions, consumer demand and a number of other inputs which affect the behavior of the end-use loads.

To simulate the behavior of a water heater, a multi-state load model is available [18], [19]. This model uses multiple states and state transition rules to describe the power demand at any given time in the simulation [20]. The physical processes within the water heater, such as thermostat set point, water temperature, consumer hot water usage, and thermodynamic heat flow equations are described by state models. These are combined to create a simulation which can simulate the power demand of thousands of individual water heater "agents", each with individualized characteristics and parameters. While this is highly advantageous for studying the effects of a thermostat setback or direct load control program on consumers, as the drop in water temperature can be tracked on an individual level, the simulations can be time- and labor-intensive.

3 Aggregated model

This section discusses how dynamic Bayesian networks, discussed in Section 2.1, can be used to model the aggregated behavior of water heaters with regards to power

consumption. First the dynamic Bayesian network model is presented, followed by a demonstration of its use for aggregated water heater end-use load forecasting.

3.1 Dynamic Bayesian network model

The structure of the dynamic Bayesian network model developed for this research is based on expert opinion. The expert opinion was used to define the relationships between a set of variables that influence the load energy consumption due to water heater operation in residential areas. These variables represent time, weather and appliance specific factors. It has been indicated by the GridLAB-D simulation that these factors are the most influential for the water heater load consumption and similar facts regarding most influential factors were pointed out in [9]. Specifically, the variables used to build the DBN are time of day (ToD), season, outside air temperature, solar radiation, water heater efficiency, water heater temperature set point, hot water usage and load consumption. The dynamic behavior of the DBN is established by using two time slices in the network structure as shown in Fig. 1. The data are extracted at 5-minute intervals from GridLAB-D, therefore the two-time slice BDN has the ability to capture the dynamic behavior of the simulation at a minimum temporal resolution of 5 minutes.

The variable relationships of the DBN model described above can be easily explained and make intuitive sense. First, regarding the time factors, it has been observed that during specific times of the day the water heater load power demand is greater. For example, the average person tends to use the shower in the morning hours, resulting in higher water usage and therefore higher water heater energy consumption at that time compared to the consumption at noon. The time of day also naturally relates to the variations observed in the outside air temperature and solar radiation. Seasonality also has an effect on the power demand since during the winter months, for example, if a water heater is located in unconditioned garage, then the lower air temperature leads to greater thermal energy loss across the insulation jacket, resulting in greater energy consumption. Seasonality also naturally relates to the variations observed in the outside air temperature and solar radiation. Finally, the appliance related variables, like water heater efficiency and thermostat set point have an intuitive relation to the load power demand. Lower efficiency water heaters will lose more heat into the ambient air, and over time consume more energy to maintain the temperature of the water as compared to a more efficient water heater. Also, the higher the thermostat set point, the more energy is consumed to maintain the temperature of the water due to the greater temperature gradient across the insulation jacket.

The network variables that relate to the first time slice are denoted with the numeric 1 of the node name, while the ones that relate to the second time slice with the numeric 2 of the node name, as shown in Fig. 1. The water heater efficiency and thermostat set point variables only appear in the first time slice, since they remain constant over time, during steady state operation. The time of day variable, belonging to the first time slice, has been observed to have an impact on the variables of the second time slice. This relationship is modeled by adding the appropriate arcs on the network as shown in Fig. 1.

3.2 Model usage example

Determining and validating a network structure that models aggregated water heater load behavior with adequate accuracy is not a trivial task. Multiple training and testing scenarios have been considered for the evaluation of the DBN model. However, once the model has been established, it provides the user with a very flexible tool for analysis and planning.

An example of its use is demonstrated in Fig. 1. The nodes of the trained network that are circled with a solid contour are the nodes to which evidences are set. The node circled with a dotted contour is the query node. In this example, the DBN querying process is used to determine the load demand, given the time of day, season, outside temperature, solar radiation, thermostat set point, efficiency and hot water usage. The querying node for the load is at the second time slice, while the evidence nodes are at the first time slice. This captures the notion that the distribution of a variable at a present time can be queried based on the values or distributions of variables at a time in the past.

Another example of using this DBN is presented in Fig. 2, in which case information related to the hot water usage needs to be derived. The nodes where evidence is set (circled with a solid contour) are time of day, season, outside temperature, solar radiation, thermostat set point, efficiency and load of the first time slice. The querying node is the hot water usage in the second time slice. In this example, similar to the aforementioned example, the evidences can be set in the form of a distribution if there is not enough information about the actual value. Also, the querying result provides the user with a distribution which is useful in accounting for forecasting errors. These two examples demonstrate the flexibility of usage that the trained DBN provides.

4 Results

In this section a description of the training and testing data is presented. The testing results of the trained DBN are compared to the simulated data under different scenarios of the water heater operation. Two different forecasting methods, soft and hard forecasting are considered based on the resulting probability distribution of the query node.

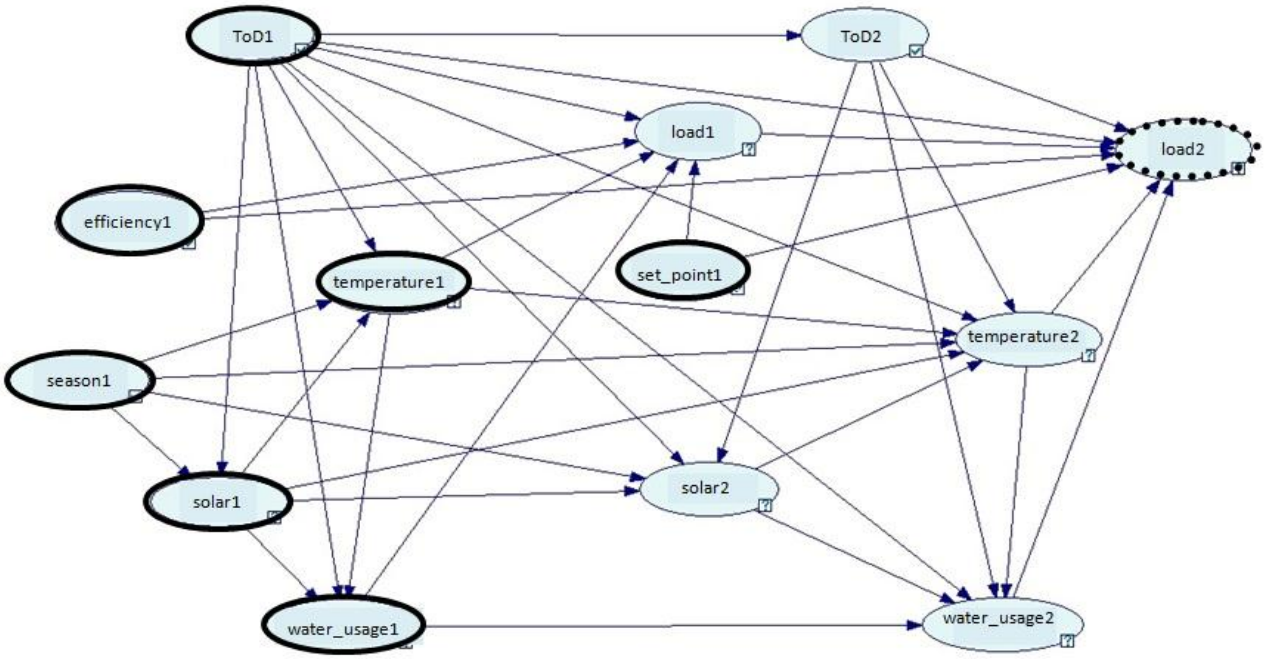


Fig. 1. Two-time slice Dynamic Bayesian Network model of aggregated water heater load. Example 1, querying load node circled with a dotted contour, using evidence information of nodes circled with solid contours.

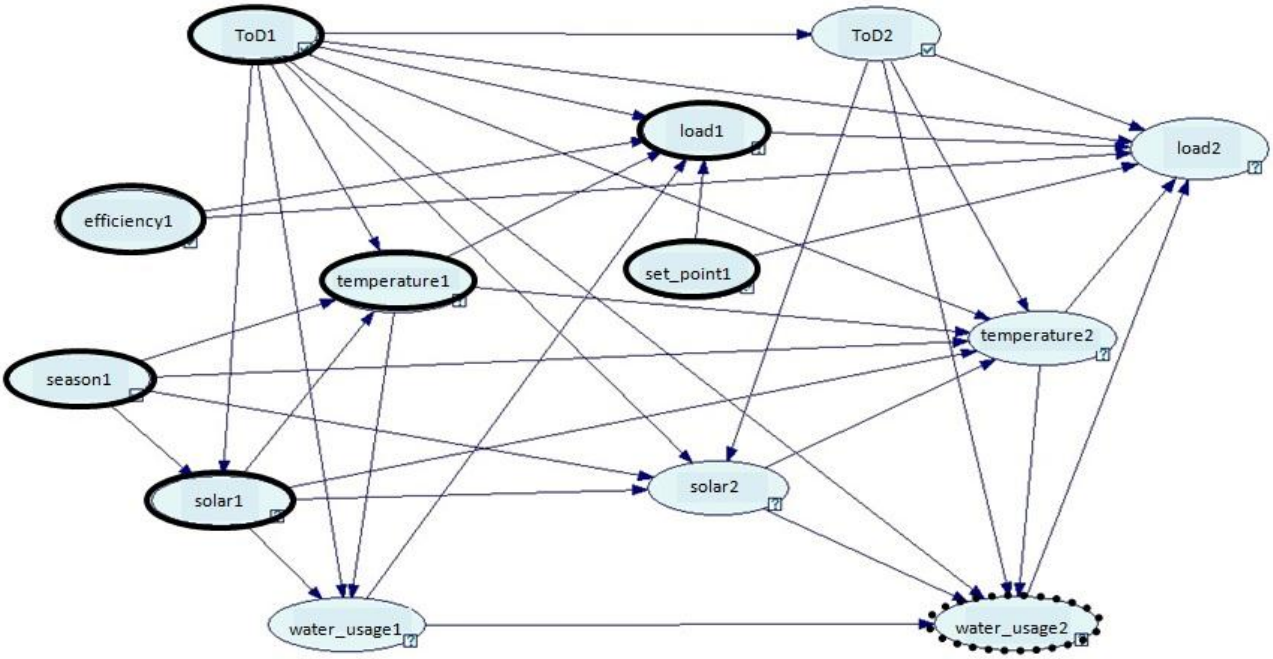


Fig. 2. Two-time slice Dynamic Bayesian Network model of aggregated water heater load. Example 2, querying water usage node circled with a dotted contour, using evidence information of nodes circled with solid contours.

4.1 Simulation data: DBN training and testing

The GridLAB-D simulation environment was used to produce the training and testing data. The simulation was of a residential neighborhood of 1000 houses. Each house had a Heating, Ventilation and Air Conditioning (HVAC) system simulated such that the inside house temperature was maintained at a reasonable level. Other characteristics, such as end-use load usage, cooling and heating set points, and thermal insulation were randomly varied across the population of homes to create a distribution of home parameters and characteristics representative of "real" building stock.

The training data were produced by running the simulation for the winter season, from December to March approximately, excluding a week in February whose data would be used for the testing dataset. This training range of data has been empirically proven to provide adequate training for the DBN. The DBN was trained using different training scenarios with the thermostat set point, efficiency and hot water usage varying between the scenarios. The water heater set point range considered was 110 to 135 ^{o}F. The water heater efficiency was set to low, medium or high, used to represent the relative amount of insulation around the thermal jacket of the water heater. Schedules (ToD) for the hot water usage were created from End-Use Load and Consumer Assessment Program (ELCAP) residential load data, while incorporating Energy Information Administration (EIA) website data on average hot water consumption in the U.S. [21],[22]. The hot water usage was set to either low or high, affecting the relative magnitude of the water flows, and was used to represent residences with low-flow rate fixtures versus older, high-flow fixtures. The simulation used a typical meteorological year (TMY) weather file that provides the outside air temperature and solar radiation information [23]. The load demand ranged from 0 to 1400 kW, approximately, between different scenarios and Time of Day.

It is a well-known fact that the discretization of the variables has a big impact on the accuracy of the querying results of the DBN [24]. The discretization method was decided based on expert opinion and experimentation with the DBN. The expert's opinion helped identify the variables with the highest sensitivity and those variables were discretized at a finer resolution. For example, the time of day is a variable with high sensitivity so it was discretized at an hourly basis. The load power demand is also an influential variable so it was uniformly discretized every 100kW. The hot water usage variable discretization is coarse, since it was only set to high or low, even if it is a highly sensitive variable. The reason for this discrepancy is that real world data do not usually contain hot water usage information with high accuracy. Future implementation of this work will involve training and testing with real world data. It is therefore appropriate to keep in consideration the realistic availability of data for a smooth transition to the real world application.

4.2 Results

The DBN was tested using GridLAB-D simulated data for a week in February that were not included in the training set. The testing of the DBN accuracy was performed by querying the load variable of the second time slice, similarly to the first example presented in Section 3.2. This example is a good indication of how a utility company would use this tool to do end-use load forecasting of an aggregated water heater load. The query results over the time period of a day, in comparison to the actual simulated data, are plotted in Fig. 3-6. The results presented in Fig. 3 correspond to a high hot water usage case and low efficiency, while the results presented in Fig. 4 correspond to low hot water usage and high efficiency. In both cases the water heater set point is set to 115 ^{o}F. Equivalently, the results in Fig. 5 and Fig. 6 present the same comparison of high water usage/low efficiency versus low water usage/high efficiency, but with the water heater set point at 130 ^{o}F. The GridLAB-D hourly average load demand data are compared to a hard and soft load demand forecasting. The hard forecasted data are obtained by selecting the load variable value that was assigned the highest probability of occurrence by the querying process. The soft forecasted data are obtained by evaluating an average of the possible load demand values weighted by their assigned probabilities.

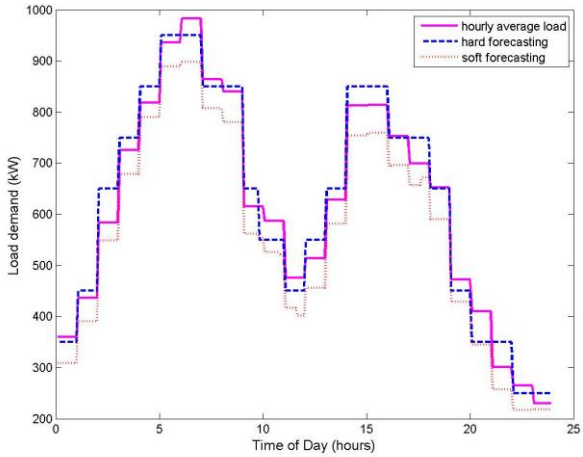


Fig. 3. Simulated vs. forecasted hourly averaged daily load profile curves. Simulation parameters: winter season, 115 ^{o}F set point, low efficiency, high water usage.

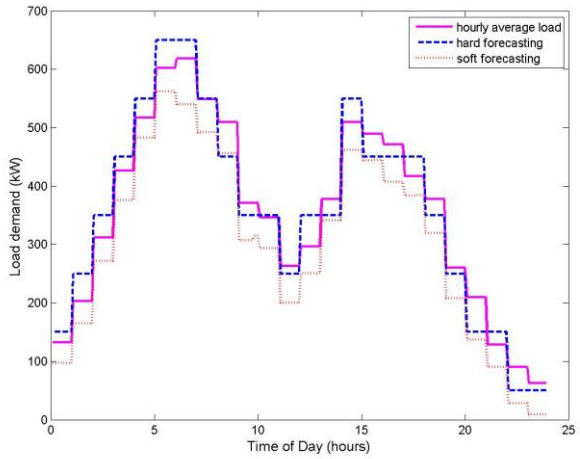


Fig. 4. Simulated vs. forecasted hourly averaged daily load profile curves. Simulation parameters: winter season, 115 ^{o}F set point, high efficiency, low water usage.

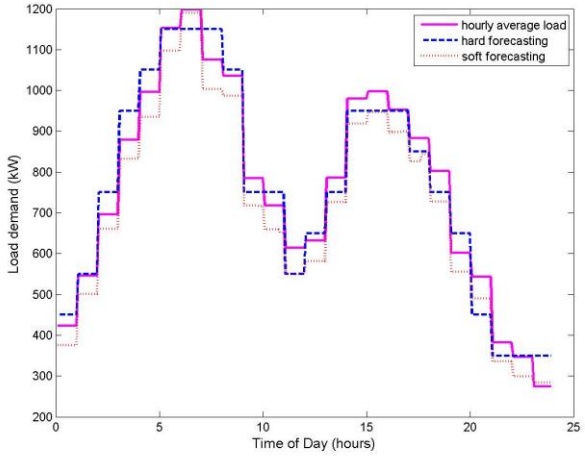


Fig. 5. Simulated vs. forecasted hourly averaged daily load profile curves. Simulation parameters: winter season, 130 ^{o}F set point, low efficiency, high water usage.

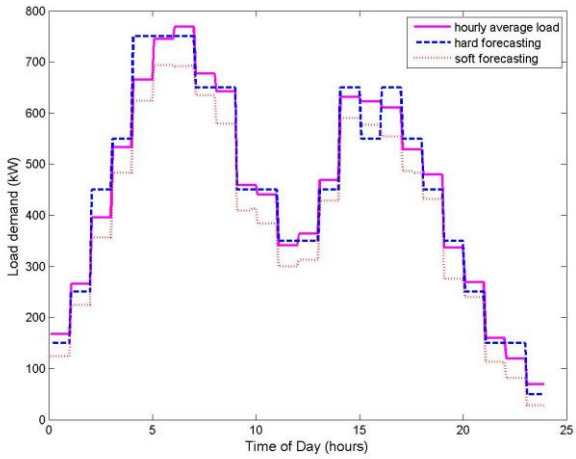


Fig. 6. Simulated vs. forecasted hourly averaged daily load profile curves. Simulation parameters: winter season, 130 ^{o}F set point, high efficiency, low water usage.

In Fig. 3 the load demand is much higher at any time in the day compared to Fig. 4, since the simulated testing data are that of a neighborhood having low efficiency water heaters and high hot water usage. Equivalently, the same statement can be made when comparing the load demand profile presented in Fig. 5 versus Fig. 6. As demonstrated by the results, the load demand additionally depends on the water heater set point. A higher set point results in higher load energy consumption as shown by comparing Fig. 3 and Fig. 5.

Both hard and soft forecasting accurately track the GridLAB-D load profile curve. It can be observed that the soft forecasting tracks the load curve variations slightly better than the hard forecasting. The average forecasting error is approximately in the order of 50kW. These results demonstrate that the DBN has been trained adequately for load forecasting of different simulated test scenarios.

5 Conclusions and future work

It has been shown that DBNs can be successfully used to model the aggregated water heater load demand, tracking the simulated data load profile curve closely. Therefore, this research provides a first indication that DBNs constitute a powerful modeling tool as applied in the area of power engineering and the smart grid. It provides the flexibility needed for energy consumption analysis and could potentially be used for the assessment of the impact of DR programs on the grid operation. It can be used to ingest a high volume of data for training under different scenarios of operation without having to modify its structure. The DBN modeling approach's main advantage over other machine learning and data mining methodologies is that it models the physical relationship between the actual system variables. The authors are now working on applying the model presented in this paper to real world data. It is expected the new results will

demonstrate the value of applying DBNs for load forecasting even further.

6 References

[1] H. Farhangi, "The path of the smart grid," IEEE PES Magazine, vol. 8, no. 1, pp. 18-28, Jan-Feb 2010.

[2] "A National Assessment of Demand Response Potential", Staff Report, Federal Energy Regulatory Commission (FERC), June 2009. [Online]. Available: http://www.ferc.gov/legal/staff-reports/06-09-demand-response.pdf

[3] M. H. Albadi and E. F. El-Saadany, "Demand Response in Electricity Markets: An Overview," IEEE PES General Meeting, 2007.

[4] P. Cappers, C. Goldman, and D. Kathan, "Demand Response in U.S. Electricity Markets: Empirical Evidence," Energy, vol. 35, pp. 1526-1535, 2010.

[5] A. Faruqui and S. Sergici. (2010, February). Household Response to Dynamic Pricing of Electricity-A Survey of the Empirical Evidence. [Online]. Available: http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1134132

[6] L. Goel, Q. Wu, and P. Wang, "Reliability Enhancement of A Deregulated Power System Considering Demand Response," IEEE PES General Meeting, 2006.

[7] R. Azami and A. F. Fard, "Impact of Demand Response Programs on System and Nodal Reliability of a Deregulated Power," IEEE Int. Conf. of Sustainable Energy Technologies, 2008.

[8] K. Moslehi and R. Kumar, "A reliability Perspective of the Smart Grid," IEEE Trans. Smart Grid, vol. 1, pp. 57-64, June 2010.

[9] R. Weron, Modeling and Forecasting Electricity Loads and Prices: A Statistical Approach. John Wiley & Sons, 2007.

[10] W. Zhang, K. Kalsi, J. Fuller, M. Elizondo, and D. Chassin, "Aggregate Model for Heterogeneous Thermostatically Controlled Loads with Demand Response," to appear in proceedings of IEEE PES General Meeting, San Diego, CA, July 2012.

[11] K. Kalsi, M. Elizondo, J. Fuller, S. Lu, and D. Chassin, "Development and Validation of Aggregated Models for Thermostatic Controlled Loads with Demand Response", Hawaii International Conference on System Sciences, Maui, Hawaii, January 2012.

[12] K. Kalsi, F. Chassin, and D. Chassin, "Aggregated Modeling of Thermostatic Loads in Demand Response: A Systems and Control Perspective", IEEE Conference on Decision and Control and European Control Conference, Orlando, Florida, December, 2011.

[13] O. Pourret, P. Naim, and B. Marcot, Bayesian Networks, A Practical Guide to Applications, Wiley, 2008.

[14] G. Zwieg and S. Russell, "Speech Recognition with Dynamic Bayesian Networks", Proc. of AAAI-98, Madison, WI, July 1998.

[15] G. Chin Jr., S. Choudhury, L. Kangas, S. McFarlane, and A. Marquez, "Fault Detection in Distributed Climate Sensor Networks using Dynamic Bayesian Networks", Proc. of 6th IEEE International Conference on e-Science, Brisbane, Australia, December 2010.

[16] D. Koller and N. Friedman, Probabilistic Graphical Models: Principles and Techniques, MIT Press, 2009.

[17] "GridLAB-D, ver. 2.2". October 2011. [Online]. Available: http://www.gridlabd.org

[18] K. P. Schneider, J. C. Fuller, and D. P. Chassin, "Multi-State Load Models for Distribution System Analysis," IEEE Trans. Power Systems, vol. 26, no. 4, pp. 2425-2433, Nov. 2011.

[19] Z. T. Taylor, K. Gowri, and S. Katipamula, "GridLAB-D Technical Support Document: Residential End-Use Module Version 1.0," PNNL-17694, Pacific Northwest National Laboratory, Richland, WA, 2008.

[20] J. C. Laurent and R. P. Malhame, "A physically-based computer model of aggregate electric water heating loads," IEEE Trans. Power Systems, vol. 9, no. 3, pp. 1209-1217, Aug. 1994.

[21] R. G. Pratt, C. C. Conner, E. E. Richman, K. G. Ritland, W. F. Sandusky, and M. E. Taylor, "Description of Electric Energy Use in Single Family Residences in the Pacific Northwest," DOE/BP 13795 21, Bonneville Power Administration, Portland, OR, 1989.

[22] "U.S. Energy Information Administration". September 2011. [Online]. Available: http://www.eia.gov.

[23] "National Solar Radiation Data Base". September 2011. [Online]. Available: http://rredc.nrel.gov/solar/old_data/nsrdb/1961-1990/tmy2/

[24] N. Friedman and M. Goldszmidt, "Discretizing Continuous Attributes While Learning Bayesian Networks", Proc. 13th International Conference on Machine Learning, vol. 159, no. 12, pp. 157-165.

Integrating Decision Tree and K-Means Clustering with Different Initial Centroid Selection Methods in the Diagnosis of Heart Disease Patients

Mai Shouman, Tim Turner, Rob Stocker
School of Engineering and Information Technology
University of New South Wales at the Australian Defence Force Academy
Northcott Drive, Canberra ACT 2600
mai.shouman@student.adfa.edu.au, t.turner@adfa.edu.au, r.stocker@adfa.edu.au

Abstract—*Heart disease is the leading cause of death in the world over the past 10 years. Researchers have been using several data mining techniques to help health care professionals in the diagnosis of heart disease patients. Decision Tree is one of the data mining techniques used in the diagnosis of heart disease showing considerable success. K-means clustering is one of the most popular clustering techniques; however initial centroid selection strongly affects its results. This paper investigates integrating k-means clustering with decision tree in the diagnosis of heart disease patients. It also investigates different methods of initial centroid selection of the k-means clustering such as inlier, outlier, range, random attribute values, and random row methods in the diagnosis of heart disease patients. The results show that integrating k-means clustering with decision tree with different initial centroid selection could enhance the decision tree accuracy in the diagnosing heart disease patients. It also showed that the inlier initial centroid selection method could achieve higher accuracy than other initial centroid selection methods in the diagnosis of heart disease patients.*

Keywords-Data Mining, K-Means Clustering, Initial Centroid Selection Methods, Decision Tree, Heart Disease Diagnosis.

1. INTRODUCTION

Heart disease is the leading cause of death in the world over the past 10 years. Moreover, the World Health Organization has reported that heart disease is the first leading cause of death in both high and low income countries [1]. The European Public Health Alliance reports that heart attacks and other circulatory diseases account for 41% of all deaths [2]. The Economical and Social Commission of Asia and the Pacific found that in one fifth of Asian countries, most lives are lost to non-communicable diseases such as cardiovascular, cancers, and diabetes diseases [3]. Statistics of South Africa report that heart and circulatory system diseases are the third leading cause of death in Africa [4]. The Australian Bureau of Statistics reported that heart and circulatory system diseases are the first leading cause of death in Australia, causing 33.7% of all deaths [5].

Motivated by the world-wide increasing mortality of heart disease patients each year and the availability of huge amount of patients' data that could be used to extract useful knowledge, researchers have been using data mining techniques to help health care professionals in the diagnosis of heart disease [6-7]. Data mining is an essential step in knowledge discovery. It is the exploration of large datasets to extract hidden and previously unknown patterns, relationships and knowledge that are difficult to detect with traditional statistical methods [8-12]. The application of data mining is rapidly spreading in a wide range of sectors such as analysis of organic compounds, financial forecasting, healthcare and weather forecasting [13].

Data mining in healthcare is an emerging field of high importance for providing prognosis and a deeper understanding of medical data. Healthcare data mining attempts to solve real world health problems in the diagnosis and treatment of diseases [14]. Researchers are using data mining techniques in the medical diagnosis of several diseases such as diabetes [15], stroke [16], cancer [17], and heart disease [18]. Several data mining techniques are used in the diagnosis of heart disease such as naïve bayes, decision tree, neural network, kernel density, bagging algorithm, and support vector machine showing different levels of accuracies [18-24]

Decision Tree is one of the successful data mining techniques used in the diagnosis of heart disease patients [19, 22, 25]. Although researchers are investigating enhancing decision tree performance in classification problems, less research is done on enhancing decision tree performance in disease diagnosis. This research investigates enhancing decision tree performance in the diagnosis of heart disease patients through integrating clustering as a preprocessing step to decision tree classification.

K-means clustering is one of the most popular and well know clustering techniques. Its simplicity and reliable behavior made it popular in many applications [26]. Initial centroid selection is a critical issue in k-means clustering and strongly affects its results [27]. This paper investigates integrating k-means clustering using different initial centroid selection methods with decision tree in the diagnosis of heart disease patients. The rest of the paper is divided as follows: the background section investigates applying data mining techniques in the diagnosis of heart disease; the

methodology section explains k-means clustering, different initial centroid selection methods, and decision tree used in the diagnosis of heart disease patients; the heart disease data section explains the data used; the results section presents the results of integrating k-means clustering and decision tree; followed by the summary section.

2. BACKGROUND

Researchers have been investigating the use of statistical analysis and data mining techniques to help healthcare professionals in the diagnosis of heart disease. Statistical analysis has identified the risk factors associated with heart disease to be age, blood pressure, smoking [28], cholesterol [29], diabetes [30], hypertension, family history of heart disease [31], obesity, and lack of physical activity [32]. Knowledge of the risk factors associated with heart disease helps health care professionals to identify patients at high risk of having heart disease.

Researchers have been applying different data mining techniques over different heart disease datasets to help health care professionals in the diagnosis of heart disease [18-19, 22-25]. The results of the different data mining research cannot be compared because they have used different datasets. However, over time a benchmark data set has arisen in the literature: the Cleveland Heart Disease Dataset (CHDD).

Decision tree is one of the data mining techniques showing considerable success compared to other data mining techniques over different heart disease datasets [19, 21-22, 25]. Applying decision tree in diagnosing heart disease patients showed different accuracies on different datasets that ranged between 60.4% and 94.93%[22, 33]. Tu et al. applied decision tree classifier on the Cleveland heart disease dataset showing accuracy of 78.9% [25].

Recently researchers are investigating enhancing decision tree performance in classification problems. Anbarasi et al. investigated enhancing decision tree performance through integrating genetic algorithm as a feature subset selection method in the diagnosis of heart disease patients [34]. This paper investigates enhancing decision tree performance in the diagnosis of heart disease patients through the integration of k-means clustering.

This paper investigates if integrating k-means clustering with decision tree can enhance the classifier's performance in diagnosing heart disease patients. Importantly, the research involves a systematic investigation of which initial centroid selection method can provide better performance in diagnosing heart disease patients. It also investigates if applying different numbers of clusters can provide different performance in diagnosing heart disease patients and which number of clusters will provide the better performance.

3. METHODOLOGY

The methodology section discusses k-means clustering with five initial centroid selection methods. It also discusses the Decision Tree classifier used in the diagnosis of heart disease patients (Figure 1).

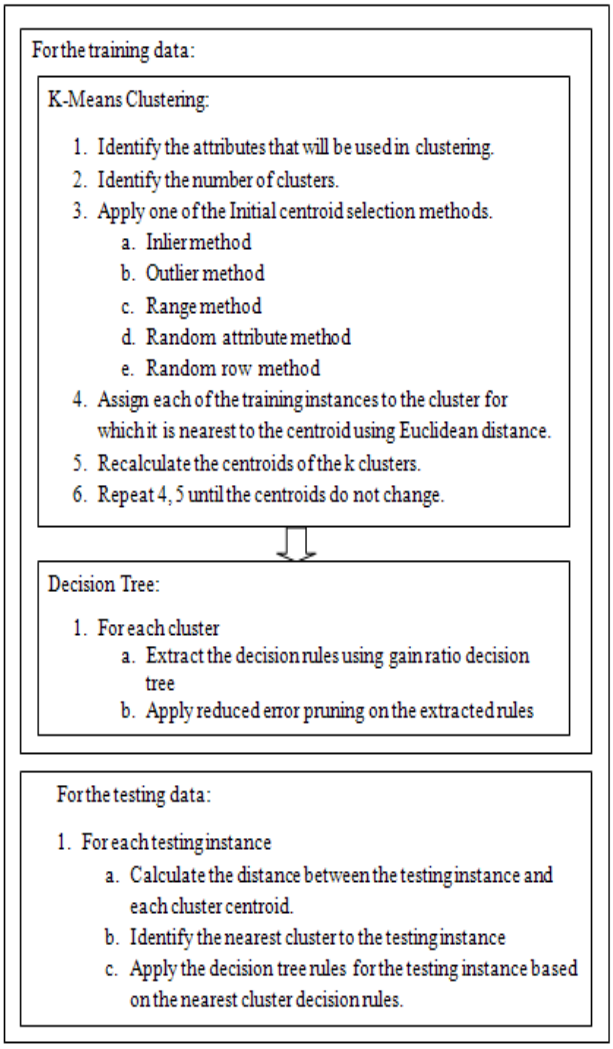


Figure 1: Integrating K-means Clustering and Decision Tree

3.1 Discretization

Decision Tree cannot deal with continuous attributes so they need to be converted into discrete ones, a process called discretization. Dougherty et al. carried out a comparative study between two unsupervised and two supervised discretization methods using 16 data sets showing that differences between the classification accuracies achieved by different discretization methods are not statistically significant [35]. Equal frequency discretization is a popular and successful unsupervised discretization method [36]. Previous related research has shown that this discretization method provides marginally better accuracy when applied on the CHHD [37]. So it is used as a preprocessing step to convert the continuous heart disease attributes to discrete ones.

3.2 K-Means Clustering

K-means clustering is one of the most popular and well know clustering techniques because of its simplicity and good behavior in many applications [26, 36]. The steps used in k-means clustering are shown in Figure 1.

Several researchers have identified that age, blood pressure and cholesterol are critical risk factors associated with heart disease [28, 31-32] . In identifying the attributes that will be used in the clustering, these attributes are obvious clustering attributes for heart disease patients. The number of clusters used in the k-means in this investigation ranged between two and five clusters. The difference between the initial centroid methods is discussed in the following section.

3.3 Initial Centroid Selection

Initial centroid selection is an important matter in k-means clustering and strongly affects its results [27]. This section discusses the generation of initial centroids based on actual sample data points using inlier method, outlier method, range method, random attribute method, and random row method [38].

3.3.1 Inlier Method

In generating the initial K centroids using the inlier method the following equations are used:

$$Ci = Min(X) - i \quad \text{where } 0 \le i \le k \qquad (1)$$

$$Cj = Min(Y) - j \quad \text{where } 0 \le j \le k \qquad (2)$$

Where the initial centroid is C (ci, cj) and min (X) and min (Y) are the minimum value of attribute X, and attribute Y respectively. K represents the number of clusters.

3.3.2 Outlier Method

In generating the initial K centroids using the outlier method the following equations are used:

$$Ci = Max(X) - i \quad \text{where } 0 \le i \le k \qquad (3)$$

$$Cj = Max(Y) - j \quad \text{where } 0 \le j \le k \qquad (4)$$

Where the initial centroid is C (ci, cj) and max (X) and max (Y) are the maximum value of attribute X, and attribute Y respectively. K represents the number of clusters.

3.3.3 Range Method

In generating the initial K centroids using the range method the following equations are used:

$$Ci = ((Max(X) - Min(X)) / K) * n \quad \text{where } 0 \le i \le k \qquad (5)$$

$$Cj = ((Max(Y) - Min(Y)) / K) * n \quad \text{where } 0 \le j \le \qquad (6)$$

The initial centroid is C (ci, cj). Where max (X) and min (X) are maximum and minimum values of attribute X, max (Y) and min (Y) are maximum and minimum values of attribute Y respectively. K represents the number of clusters.

3.3.4 Random Attribute Method

In generating the initial K centroids using the random attribute method the following equations are used:

$$Ci = random(X) \quad \text{where } 1 \le i \le k \qquad (7)$$

$$Cj = random(Y) \quad \text{where } 1 \le j \le k \qquad (8)$$

The initial centroid is C (ci, cj). The values of 'i', and 'j' vary from 1 to 'k'.

3.3.5 Random Row Method

In generating the initial K centroids using the random row method the following equations are used:

$$I = random(V) \quad \text{where } 1 \le V \le N \qquad (9)$$

$$Ci = X(I) \qquad (10)$$

$$Cj = Y(I) \qquad (11)$$

The initial centroid is C (ci, cj). N is the number of instances in the training dataset. X (I) and Y(I) are the values of the attributes X and Y respectively for the instance I.

3.4 Decision Tree

The decision tree type used in this research is the gain ratio decision tree. The gain ratio decision tree is based on the entropy (information gain) approach, which selects the splitting attribute that minimizes the value of entropy, thus maximizing the information gain [36]. To identify the splitting attribute of the decision tree, one must calculate the information gain for each attribute and then select the attribute that maximizes the information gain. The information gain for each attribute is calculated using the following formula [8, 36]:

$$E = \sum_{i=1}^{k} Pi \log_2 Pi \qquad (12)$$

Where k is the number of classes of the target attribute

Pi is the number of occurrences of class i divided by the total number of instances (i.e. the probability of i occurring).

To reduce the effect of bias resulting from the use of information gain, a variant known as gain ratio was introduced by the Australian academic Ross Quinlan [36]. The information gain measure is biased toward tests with many outcomes. That is, it prefers to select attributes having a large number of values [8]. Gain Ratio adjusts the information gain for each attribute to allow for the breadth and uniformity of the attribute values.

Gain Ratio = Information Gain / Split Information (13)

Where the split information is a value based on the column sums of the frequency table [36].

After extracting the decision tree rules, reduced error pruning was used to prune the extracted decision rules. Reduced error pruning is one of the fastest pruning methods and known to produce both accurate and small decision rules [39]. Applying reduced error pruning provides more compact decision rules and reduces the number of extracted rules.

3.5 10 Fold Cross Validation

To measure the stability of the proposed model, the data is divided into training and testing data with 10-fold cross validation. To evaluate the performance of the proposed model the sensitivity, specificity, and accuracy are calculated. The sensitivity is the proportion of positive instances that are correctly classified as positive (e.g. the proportion of sick people that are classified as sick). The specificity is the proportion of negative instances that are correctly classified as negative (e.g. the proportion of healthy people that are classified as healthy). The accuracy is the proportion of instances that are correctly classified [36].

Sensitivity = True Positive / Positive (14)

Specificity = True Negative / Negative (15)

Accuracy = (True Positive + True Negative) / (Positive + Negative) (16)

4. HEART DISEASE DATA

The data used in this study is the Cleveland Clinic Foundation Heart disease data set available at http://archive.ics.uci.edu/ml/datasets/Heart+Disease. The data set has 76 raw attributes. However, all of the published experiments only refer to 13 of them. The data set contains 303 rows of which 297 are complete. Six rows contain missing values and they are removed from the experiment. The attributes used in this study are shown in Table 1.

Table 1: Selected Cleveland Heart Disease Data Set Attributes

Name	Type	Description
Age	Continuous	Age in years
Sex	Discrete	1 = male 0 = female
Cp	Discrete	Chest pain type: 1 = typical angina 2 = atypical angina 3 = non-anginal pain 4 =asymptomatic
Trestbps	Continuous	Resting blood pressure (in mm Hg)
Chol	Continuous	Serum cholesterol in mg/dl
Fbs	Discrete	Fasting blood sugar > 120 mg/dl: 1 = true 0 = false
Restecg	Discrete	Resting electrocardiographic results: 0 = normal 1 = having ST-T wave abnormality 2 =showing probable or define left ventricular hypertrophy
Thalach	Continuous	Maximum heart rate achieved
Exang	Discrete	Exercise induced angina: 1 = yes 0 = no
Old peak ST	Continuous	Depression induced by exercise relative to rest
Slope	Discrete	The slope of the peak exercise segment : 1 = up sloping 2 = flat 3= down sloping
Ca	Discrete	Number of major vessels colored by fluoroscopy that ranged between 0 and 3.
Thal	Discrete	3 = normal 6= fixed defect 7= reversible defect
Diagnosis	Discrete	Diagnosis classes: 0 = healthy 1= patient who is subject to possible heart disease

5. RESULTS

A range of single and combined number of clustering attributes is applied in the experiment involving age, blood pressure and cholesterol attributes. However, best results are found using single attribute which is the age attribute. So K-means clustering is applied using the age attribute then the decision tree is applied on the thirteen attributes. The results of sensitivity, specificity, and accuracy in the diagnosis of heart disease using k-means clustering and decision tree with different initial centroids selection methods and different numbers of clusters are shown in Table 2. For the random attribute and random row methods, ten runs are executed and the average and best for each method are calculated and shown in Table 2. Tables 2 show that the best accuracy achieved is 83.9% by the inlier method with two clusters. The range method with different numbers of clusters did not show any enhancement in the decision tree accuracy in the diagnosis of heart disease patients.

Increasing the number of clusters with the inlier method did not show any enhancement in the accuracy as shown in Figure 2. Increasing the number of clusters with the outlier method could enhance its accuracy and showed the best accuracy with three clusters as shown in Figure 3. Increasing the number of clusters with the range method could enhance its accuracy and showed the best accuracy with four clusters as shown in Figure 4. However these accuracies are still less than that achieved by the inlier method with two clusters. Increasing the number of clusters for the random attribute and the random row could achieve slight enhancement in the

accuracy but it is still less than that achieved by them with two clusters as shown in Figure 5, and 6 respectively.

Table 2: Integrating different initial centroid selection for k-means clustering with Decision tree in diagnosing heart disease patients

No of Clusters	Initial Centroid Selection Method		Sensitivity	Specificity	Accuracy
No of clusters = 2	Inlier Method		81.6	83	83.9
	Outlier Method		71.6	76.2	76
	Range Method		71.6	76.2	76
	Random Attribute	Avg	75.85	78.85	79.29
		Best	77.7	83.3	82.2
	Random Row	Avg	76.94	79.51	80.14
		Best	81.6	83	83.9
No of clusters = 3	Inlier Method		76.6	80.2	80.9
	Outlier Method		78.1	79.6	81.2
	Range Method		69.8	77.8	76.3
	Random Attribute	Avg	73.17	78.07	78.33
		Best	76	79.9	80.3
	Random Row	Avg	72.71	78.06	77.95
		Best	76.2	78.9	79.8
No of clusters = 4	Inlier Method		72.8	80	78.5
	Outlier Method		74.1	80.3	79.9
	Range Method		72.8	80	78.5
	Random Attribute	Avg	72.31	79.05	78.01
		Best	74.2	80.7	80.5
	Random Row	Avg	72.07	79.14	78.05
		Best	72.3	81.1	79.9
No of clusters = 5	Inlier Method		78.2	77.2	75.9
	Outlier Method		68.1	72	73.6
	Range Method		72.8	77.2	75.9
	Random Attribute	Avg	71.59	76.2	75.56
		Best	73.5	77.3	78.1
	Random Row	Avg	72.3	75.86	75.6
		Best	76.7	75	78

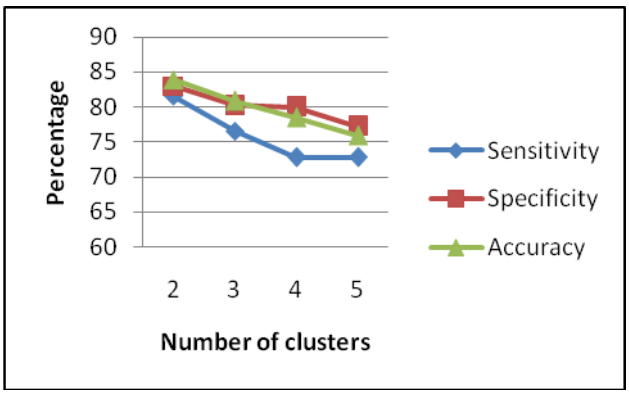


Figure 2: Different Number of Clusters Performance for Inlier Method

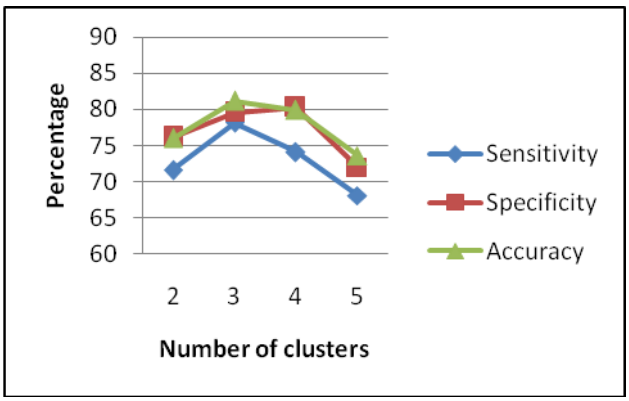


Figure 3: Different Number of Clusters Performance for Outlier Method

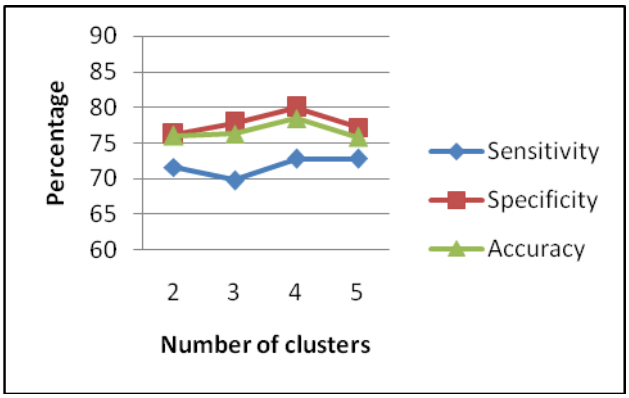


Figure 4: Different Number of Clusters Performance for Range Method

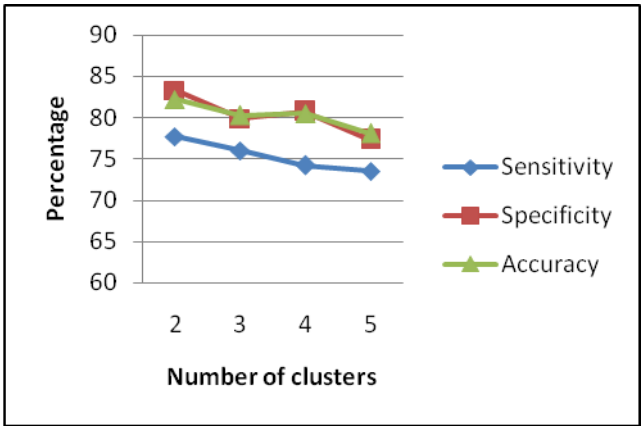


Figure 5: Different Number of Clusters Performance for Random Attribute Method

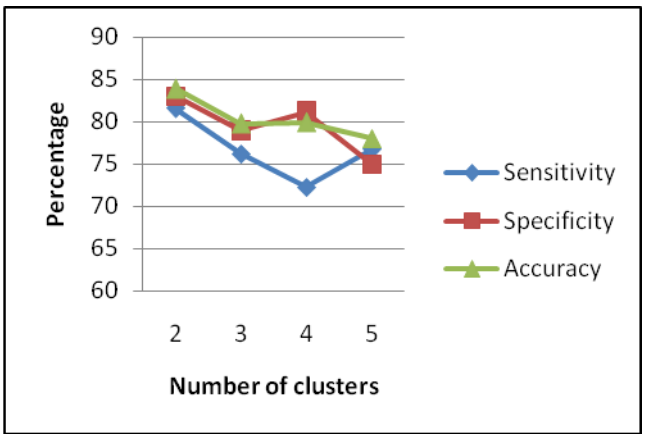


Figure 6: Different Number of Clusters Performance for Random Row Method

Why do two clusters show better performance than other numbers of clusters in the diagnosis of heart disease patients? The number of instances is relatively small in the CHHD. A larger dataset is needed to identify if two clusters will still provide the best results. Also, the target attribute of the Cleveland heart disease dataset has two values. Further investigation is also needed to identify if there is a relationship between the number of clusters showing best results and the number of values of the target attribute.

When comparing integrating k-means clustering and decision tree with traditional decision tree applied previously on the same dataset, integrating k-means clustering with decision tree could enhance the accuracy of decision tree in diagnosing heart disease patients as shown in Table 3. In Addition, integrating k-means clustering and decision tree could achieve higher accuracy than the bagging algorithm in the diagnosis of heart disease patients as shown in Table 3.

Table 3: Comparing integrating k-means clustering and decision tree with traditional decision tree and other data mining techniques

Author/ Year	Technique	Accuracy
Tu, et al., 2009	Decision tree	78.91%
	Bagging Algorithm	81.41%
Our work	Two clusters Inlier initial centroid selection k-means clustering decision tree	83.9%

6. SUMMARY

Heart disease is the leading cause of death all over the world. Researchers have been investigating applying different data mining techniques to help health care professionals in the diagnosis of heart disease patients. Decision Tree is one of the successful data mining techniques used in the diagnosis of heart disease patients. This paper investigated integrating k-means clustering with decision tree in the diagnosis of heart disease patients. Initial centroid selection is a critical issue that strongly affects k-means clustering results. Our research systematically investigated applying different methods of initial centroid selection such as range, inlier, outlier, random attribute values, and random row methods for the k-means clustering technique in the diagnosis of heart disease patients. The results show that integrating k-means clustering and decision tree can enhance decision tree accuracy in the diagnosis of heart disease patients. The results also show that the best accuracy achieved is 83.9% by the inlier method with two clusters. Finally, some limitations on this work are noted as pointers for future research.

7. REFERENCES

[1] World Health Organization. 2007 7-Febuary 2011]; Available from: http://www.who.int/mediacentre/factsheets/fs310.pdf.

[2] European Public Health Alliance. 2010 7-February-2011]; Available from: http://www.epha.org/a/2352

[3] ESCAP. 2010 7-February-2011]; Available from: http://www.unescap.org/stat/data/syb2009/9.Health-risks-causes-of-death.asp.

[4] Statistics South Africa. 2008 7-February-2011]; Available from: http://www.statssa.gov.za/publications/P03093/P030932006.pdf

[5] Australian Bureau of Statistics. 2010 7-February-2011]; Available from: http://www.ausstats.abs.gov.au/Ausstats/subscriber.nsf/0/E8510D1C8DC1AE1CCA2576F600139288/$File/33030_2008.pdf

[6] Helma, C., E. Gottmann, and S. Kramer, Knowledge discovery and data mining in toxicology. Statistical Methods in Medical Research, 2000.

[7] Podgorelec, V., et al., Decision Trees: An Overview and Their Use in Medicine. Journal of Medical Systems, 2002. Vol. 26.

[8] Han, j. and M. Kamber, Data Mining Concepts and Techniques. 2006: Morgan Kaufmann Publishers.

[9] Lee, I.-N., S.-C. Liao, and M. Embrechts, Data mining techniques applied to medical information. Med. inform, 2000.

[10] Obenshain, M.K., Application of Data Mining Techniques to Healthcare Data. Infection Control and Hospital Epidemiology, 2004.

[11] Sandhya, J., et al., Classification of Neurodegenerative Disorders Based on Major Risk Factors Employing Machine Learning Techniques. International Journal of Engineering and Technology, 2010. Vol.2, No.4.

[12] Thuraisingham, B., A Primer for Understanding and Applying Data Mining. IT Professional IEEE, 2000.

[13] Ashby, D. and A. Smith, The Best Medicine? Plus Magazine - Living Mathematics., 2005.

[14] Liao, S.-C. and I.-N. Lee, Appropriate medical data categorization for data mining classification techniques. MED. INFORM., 2002. Vol. 27, no. 1, 59–67, .

[15] Porter, T. and B. Green, Identifying Diabetic Patients: A Data Mining Approach. Americas Conference on Information Systems, 2009.
[16] Panzarasa, S., et al., Data mining techniques for analyzing stroke care processes. Proceedings of the 13th World Congress on Medical Informatics, 2010.
[17] Li L, T.H., Wu Z, Gong J, Gruidl M, Zou J, Tockman M, Clark RA, Data mining techniques for cancer detection using serum proteomic profiling. Artificial Intelligence in Medicine, Elsevier, 2004.
[18] Das, R., I. Turkoglu, and A. Sengur, Effective diagnosis of heart disease through neural networks ensembles. Expert Systems with Applications, Elsevier, 2009. 36 (2009): p. 7675–7680.
[19] Andreeva, P., Data Modelling and Specific Rule Generation via Data Mining Techniques. International Conference on Computer Systems and Technologies - CompSysTech, 2006.
[20] Hara, A. and T. Ichimura, Data Mining by Soft Computing Methods for The Coronary Heart Disease Database. Fourth International Workshop on Computational Intelligence & Applications, IEEE, 2008.
[21] Rajkumar, A. and G.S. Reena, Diagnosis Of Heart Disease Using Datamining Algorithm. Global Journal of Computer Science and Technology, 2010. Vol. 10 (Issue 10).
[22] Sitar-Taut, V.A., et al., Using machine learning algorithms in cardiovascular disease risk evaluation. Journal of Applied Computer Science & Mathematics, 2009.
[23] Srinivas, K., B.K. Rani, and A. Govrdhan, Applications of Data Mining Techniques in Healthcare and Prediction of Heart Attacks. International Journal on Computer Science and Engineering (IJCSE), 2010. Vol. 02, No. 02: p. 250-255.
[24] Yan, H., et al., Development of a decision support system for heart disease diagnosis using multilayer perceptron. Proceedings of the 2003 International Symposium on, 2003. vol.5: p. pp. V-709- V-712.
[25] Tu, M.C., D. Shin, and D. Shin, Effective Diagnosis of Heart Disease through Bagging Approach. Biomedical Engineering and Informatics, IEEE, 2009.
[26] Wu, X., et al., Top 10 algorithms in data mining analysis. Knowl. Inf. Syst., 2007.
[27] Tajunisha, N. and V. Saravanan, A new approach to improve the clustering accuracy using informative genes for unsupervised microarray data sets. International Journal of Advanced Science and Technology, 2011.
[28] Heller, R.F., et al., How well can we predict coronary heart disease? Findings in the United Kingdom Heart Disease Prevention Project. BRITISH MEDICAL JOURNAL, 1984.
[29] Wilson, P.W.F., et al., Prediction of Coronary Heart Disease Using Risk Factor Categories. American Heart Association Journal, 1998.
[30] Simons, L.A., et al., Risk functions for prediction of cardiovascular disease in elderly Australians: the Dubbo Study. Medical Journal of Australia, 2003. 178.
[31] Salahuddin and F. Rabbi, Statistical Analysis of Risk Factors for Cardiovascular disease in Malakand Division. Pak. j. stat. oper. res., 2006. Vol.II: p. pp49-56.
[32] Shahwan-Akl, L., Cardiovascular Disease Risk Factors among Adult Australian-Lebanese in Melbourne. International Journal of Research in Nursing, 2010. 6 (1).
[33] Palaniappan, S. and R. Awang, Web-Based Heart Disease Decision Support System using Data Mining Classification Modeling Techniques. Proceedings of iiWAS, 2007.
[34] Anbarasi, M., E. Anupriya, and N.C.S.N. Iyengar, Enhanced Prediction of Heart Disease with Feature Subset Selection using Genetic Algorithm. International Journal of Engineering Science and Technology, 2010. Vol. 2(10).
[35] Dougherty, J., R. Kohavi, and M. Sahami, Supervised and unsupervised discretization of continuous features. In: Proceedings of the 12th international conference on machine learning. San Francisco: Morgan Kaufmann, 1995: p. p. 194–202.
[36] Bramer, M., Principles of data mining. 2007: Springer.
[37] Shouman, M., T. Turner, and R. Stocker, Using decision tree for diagnosing heart disease patients. 9th Australasian Data Mining Conference 2011. 121.
[38] Khan, D.M. and N. Mohamudally, A Multiagent System (MAS) for the Generation of Initial Centroids for kmeans Clustering Data Mining Algorithm based on Actual Sample Datapoints. Journal of Next Generation Information Technology, August, 2010. Volume 1, Number 2.
[39] Esposito, F., D. Malerba, and G. Semeraro A Comparative Analysis of Methods for Pruning Decision Trees. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 1997. VOL. 19, NO. 5.

Forecasting Stock Price Movement with Semi-Supervised Learning

Kanghee Park and Hyunjung Shin*

Abstract— **Stock price prediction is a field that has been continuously interested. Stock price indexes represent correlation between the company and the others including the influence of oil prices, exchange rates, money interest rates, stock price indexes in other countries, and economic situations, the indexes are sensitively influenced by the fluctuation of these factors. To overcome the complexity, this paper proposes a network based method incorporating the relations between the stock prices and the other factors by using a graph-based semi-supervised learning algorithm. For verifying the significance of the proposed method, it was applied to the prediction problems of company stock prices listed in the KOSPI from January 2007 to August 2008.**

I. INTRODUCTION

Interests on the stock price prediction have been continued according to the public understanding in stock investment. Factors that affect stock prices are oil prices, exchange rates, interest rate, stock price indexes in other countries, and economic situations. Studies on stock price prediction methods based on these factors have been variously conducted [1-4].

Various stock price prediction methods using a time series analysis method have been presented. Jeantheau (2004) predicted stock prices using an ARCH model, and Amilon (2003) and Liu et al. (2009) proposed a prediction method using a GARCH model based on the Skewed-GED Distribution for Chinese stock markets [2-3]. As these methods perform the prediction using a time series analysis method based on the past stock price validity, an assumption in which the future stock price will be varied as similar to that of the past is a basis. The time series data obtained from some natural phenomena, such as numbers of sunspot cycles, rain falls, temperature, and others, nicely follows such an assumption. It is possible to obtain excellent results using the time series analysis method.

Although these various factors mentioned above affect stock prices directly, they have influence on the stock price indirectly through a complex interrelation between these factors. For instance, although interests rates and exchange rates directly affect stock price fluctuations, they have influence on the stock price based on the reciprocal relationship between these two factors.

The authors would like to gratefully acknowledge support from Post Brain Korea 21 and the research grant from National Research Foundation of Korean government (KRF-2010-0007804).

Kanghee Park (can17@ajou.ac.kr) is in the Ph.D course of Industrial & Engineering, Ajou University, Suwon, 443-749 Korea

* Corresponding author: Hyunjung Shin (shin@ajou.ac.kr) is a professor of the department of Industrial & Information Systems Engineering, Ajou University, Suwon, 443-749, Korea

However, there are some methodological limitations that include the relationship between these factors and reciprocal complexity to a time series model specifically and its formalization [5-6]. Also, many studies on the stock price prediction in the machine learning have been conducted. The artificial neural network (ANN) and support vector machine (SVM) methods have been frequently used as a typical model [7-9]. Tay and Cao (2001) proposed a method that introduces financial time series data to the SVM, and Kanas (2003) attempted the prediction of the S&P500 index using the ANN model [10-11]. Also, Yang et al. (2001) proposed an early warning system of commercial bank loan risks using the ANN model, and Bekiros and Georgoutsos (2008) analyzed that how uncertain news, which show a difficulty in identifying bullish and bearish factors, affect the NASDAQ index using the ANN model [12-13].

Although the methods using ANN and SVM include the interrelation and complexity between the stock price and these factors in its modeling specifically, it is still insufficient. It does not formalize the interrelation between factors even though the factors in fluctuating stock prices and their interrelation are expressed in the model [14]. For instance, ANN and SVM can represent how the fluctuations in interests, exchange rates, and oil prices affect the fluctuations in stock prices primarily. However, it is somewhat difficult to express how a drop in interests affects the exchange rates and then how these changes affect the next situation, i.e., how the second, third, and additional higher level interrelations affect the stock prices eventually. In addition, it is not easy to identify how the changes in stock prices caused by such a sequential process affect these factors again. That is, there is a limitation in presenting the complexity between factors. In this study, a stock price prediction method that uses semi-supervised learning (SSL), which has been recently attracted in the field of machine learning methods, is proposed to solve this limitation [15-16].

SSL that is one of recently developed machine learning methods is an analysis method through defining the interrelation between factors to a network [15-16]. SSL can consider the interrelation and complexity between factors through a network. It connects individual networks using the similarities between factors and extracts the influence of the final similarities in the connected input factors and responded factors as its prediction value. In this study, a stock prediction model that considers the interrelation and multi-dimensional causal complexity in various economic indexes by combining time series data to SSL is proposed. The proposed model was applied to the stock price prediction for individual companies listed to KOSPI from January 2007 to August 2008 and its performance was also verified.

This study consists of five sections. Section 2 describes the methodology of SSL. Section 3 proposes a method that combines time series data to SSL. Section 4 represents

experiments and results. Finally, Section 5 shows the conclusion of this study.

II. SEMI-SUPERVISED LEARNING (SSL)

In graph-based SSL algorithm, a data point $x_i \in R^M$ *(i* = 1,…, *n)* is represented as a node *i* in a graph, and the relationship between data points is represented by an edge where the connection strength from each node *j* to each other node *i* is encoded as w_{ij} of a weight matrix *W* [17]. Fig. 1 presents a graphical representation of SSL.

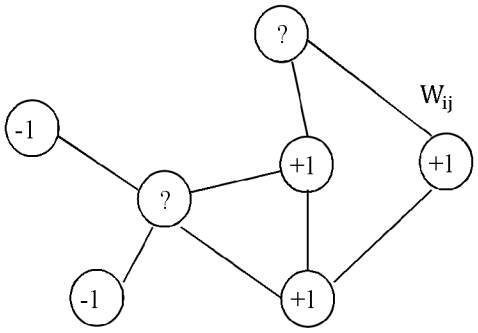


Fig. 1 Graph-based semi-supervised learning (SSL).

A weight w_{ij} can take a binary value (*0* or *1*) in the simplest case. Often, a Gaussian function of Euclidean distance between points with length scale σ is used to specify connection strength:

$$w_{ij} = \begin{cases} \exp\left(-\frac{(x_i-x_j)^T(x_i-x_j)}{\sigma^2}\right) & \text{if } i\sim j \text{ ('k' nearest neighbors)} \\ 0 & \text{otherwise} \end{cases}, \quad (1)$$

Usually, an edge $i\sim j$ is established when node *i* is one of *k*-nearest neighbors of node *j* or node *i* is within a certain Euclidean distance *r*, $\|x_i-x_j\| < r$. The labeled nodes have labels $y_l \in \{-1,1\}$ $(l = 1,\dots,L)$, while the unlabeled nodes have zeros $y_u = 0$ *(u=L+1,...,L+U)*. The algorithm will output an n-dimensional real-valued vector $f = [f_l^T\ f_u^T]^T = (f_1,\dots,f_L,f_{L+1},\dots,f_{L+U})^T$ which can be thresholded to make label predictions on $f_{L+1},\dots,f_{L+U}$ after learning. It is assumed that (a) f_i should be close to the given label y_i in labeled nodes and (b) overall, f_i should not be too different from its adjacent nodes f_j. One can obtain *f* by minimizing the following quadratic functional:

$$\mathrm{Min}_f\ (f-y)^T(f-y) + \mu f^T L f, \quad (2)$$

where $y = (y_1,\dots,y_l,0,\dots,0)^T$, and the matrix $\boldsymbol{L}$, called the graph Laplacian, is defined as $L = D - W$, $D = \mathrm{diag}(d_i)$, and $d_i = \sum_j \omega_{ij}$. The first term corresponds to the loss function in terms of condition (a), and the second term represents the smoothness of the predicted outputs in terms of condition (b). The parameter μ represents trades between loss and smoothness. The solution to (2) is obtained as

$$f = (I + \mu L)^{-1} y, \quad (3)$$

where *I* is the identity matrix.

III. PROPOSED METHOD

To apply the graph-based SSL to time series prediction, we propose a method of graph representation for time series data, and a procedure for obtaining predicted values from the graph. For instance, assume that multiple time series are given as the input for the prediction problem of the stock price of Hyundai Motors: the stock price of LG chem, the stock price of KIA Motors, WTI intermediate oil price, *etc.* To apply SSL to this problem, the proposed method begins with a re-designed graph as in Fig. 2.

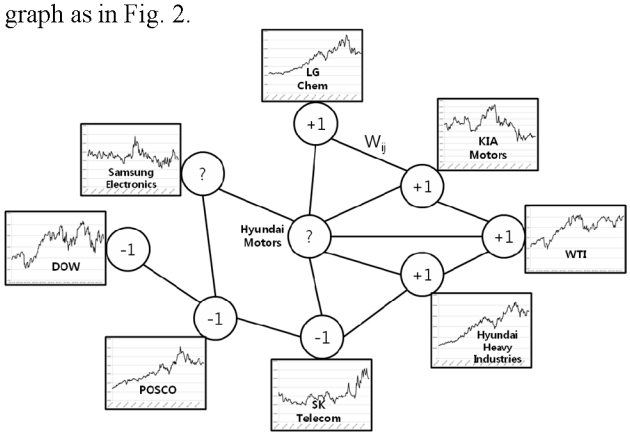


Fig. 2 Graph SSL representation for time series prediction.

The nodes in the graph represent the time series variables that influence the stock price of Hyundai Motors, e.g., the stock price of LG chem, the stock price of KIA Motors, WTI intermediate oil price and other external factors. Then the edge between any two nodes $i\sim j$ stands for the similarity of the two sets of time series, represented as '$w_{ij} \in W$'. The label 'y_t' on each node presents either 'up' (+1) or 'down' (-1) of the time series at time point *t*. In the graph of <Figure 2>, the labels of Hyundai Motors are not known yet at time point *t*, and hence are unlabeled. To estimate the label y_t, the similarity matrix of SSL was calculated at time point t-1, W_{t-1}. Based on this set-up, we explain how to measure the similarity 'w_{ij}' of a weight matrix *W* and how to set the value for label 'y'.

A. Similarity Matrix

The design of the similarity matrix *W* plays a critical part in the aspect of performance when using SSL[16, 18]. In the matrix *W*, each element represents how strongly the two nodes are related, with larger elemental value being associated with greater nodal similarity. In the proposed method, the time-series data are transformed by building technical indicators (TIs). The general process of constructing the similarity matrix is described in Fig. 2.

TIs are frequently used in financial forecasting as they offer the advantages of removing the noise (oscillatory noise) inherent in time series and illustrating the underlying structure, i.e., the tendencies and structural factors affecting variation[5-6, 19]. Stock prices and other economic indices exist as time series data by the nature of the variables, and each of them is defined as a sequence as

$$X_t = \{x_1, x_2, \dots, x_i, \dots, x_t\}, \quad (4)$$

where *t* represents the current time point, and x_t is the corresponding value. The existence of X_t as time series data induces several problems in the direct application of SSL to the data. As shown in Fig. 1, each of the nodes on the graph has its own time series, as shown in (4). For instance, the Hyundai Motors node has $X_t^{\text{Hyundai Motors}}$ and the LG chem node also has $X_t^{\text{LG chem}}$. The problem is that it is difficult to draw the similarity between them directly from the two sets of series data. Therefore, individual time series are transformed into structural characteristics of time point *t*, i.e., $S_t^{Hyundai\ Motors}$ and $S_t^{LG\ chem}$, representing the tendencies and factors for variation of individual series. Table I summarizes the TIs used in this study. The similarity between the two nodes is measured by using the seven-tuple vector $S_t = \{s_1, s_2, s_3, s_4, s_5, s_6, s_7\}$ composed of MA, BIAS, OSC, ROC, K, D, and RSI.

Using the TIs enables the time series data to be transformed into TIs-type data, while maintaining the time associations of the series, and thus eases their application to SSL.

TABLE I THE DEFINITION OF TECHNICAL INDICATORS (TIS)

	TIs	Meaning
s_1	$MA_p(X_t) = \frac{1}{p}(x_t) + \frac{p-1}{p} MA_p(X_{t-1})$	p-moving average (exponential smoothing)
s_2	$BIAS_p(X_t) = \frac{x_t - MA_p(X_t)}{MA_p(X_t)}$	The change rate of x_t relative to $MA_p(X_t)$
s_3	$OSC_{p,q}(X_t) = \frac{MA_p(X_t) - MA_q(X_t)}{MA_p(X_t)}$	The change rate of $MA_q(X_t)$ relative to $MA_p(X_t)$
s_4	$ROC_p(X_t) = \frac{x_t - x_{t-p}}{x_t}$	The relative rate of change for X_t between p consecutive time points
s_5	$K_t^p = \frac{x_t - Min_{i=t-p-1}^{t}(x_i)}{Max_{i=t-p-1}^{t}(x_i) - Min_{i=t-p-t}^{t}(x_i)}$	Standardization of x_t
s_6	$D_t^p = MA_3(K_t^p)$	3- Moving Average of K_t^p
s_7	$RSI_t^p = \frac{\sum_{i=t-p-1}^{t}(\lvert x_i - x_{i-1}\rvert)}{\sum_{i=t-p-t}^{t}(\lvert x_i - x_{i-1}\rvert)}$,	The relative strength index.

B. Label

The label on the node in the SSL graph in Fig. 1 is designed to explain whether the predicted value of the corresponding variable is thumbs-up or down. It can be formulated as follows:

$$y_t = \text{sign}(x_t - MA_5(x_t)). \quad (5)$$

For instance, if the total amount of the Hyundai Motors' stock price (t) exceeds the 5-days moving average, (5) will give a '$y_t = +1$' label. On the contrary, the node is labeled as '$y_t = -1$' for the opposite case. And '$y_t = 0$' if there is no information about the movement of the corresponding time series at time point *t*, the label is to be predicted. In the proposed method, we set the label of the target variable to '0'. Given label y_t, equation (3) provides the predicted value f_t for every node, which can take on a real number unlike the values of label y_t.

If '$f_t > 0$', it means the stock price will increase relative to the average of the MA(5), therefore one can take the position of "buy" for the stock. On the other hand, one can take the position of "sell" otherwise. This procedure is described in Fig. 3.

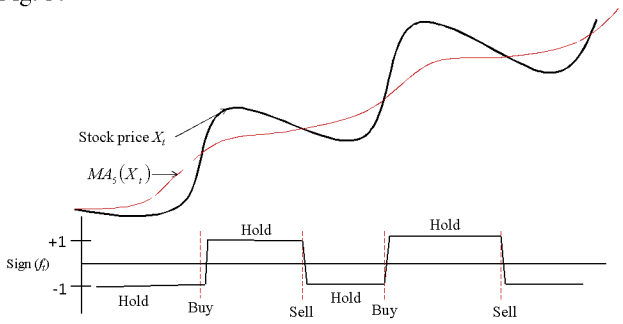


Fig. 3 Interpretation for forecasted values and simple trading strategy

IV. EXPERIMENT

A. Data

The data used in this experiment was presented by a total of 403 daily data points from January 2007 to August 2008. The factors employed as variables were the major global economic indexes, such as Dow-Jones average (DOW), National association of securities dealers automated quotations (NASDAQ), Japanese stock market index (NIKKEI), Hang seng index (HSI), Shanghai composite index (SSE), Taiwan stock exchange corporation (TSEC), Financial times security exchange (FTSE), Deutscher aktien index (DAX), continuous assisted quotation index (CAC), Bombay stock exchange portmanteau of sensitive and index (BSE_SENSEX), Indice bovespa (IBOVESPA), Australia all ordinaries index (AORD), Korea composite stock price index (KOSPI), exchange rate(KRW-USD), the west texas intermediate oil price (WTI), and the certificate of deposit (CD). Also, the stock prices of 200 companies listed to

KOSPI200 were included. Table 1-(Appendix) shows the list of these 200 companies.

B. Experimental Setting

The SSL model proposed in this experiment was compared with the ANN and SVM models. The ANN model used a multilayer perception function. The SVM model used an RBF kernel function that has been known as an excellent performance model relatively. A total of 103 daily data from January 2007 to May 2007 were determined as a training and validation period and the performance for a total of 300 daily data from June 2007 to August 2008 was compared. SSL was predicted using a rolling forecast method [20]. The rolling forecast method predicts a point of t+1 using the data from a point of 1 to a point of t and applies a learning data period from a point of 2 to the point of t+1 for the prediction of a point of t+2. In general, the ANN and SVM models represent higher performance as its learning data periods are highly determined. However, the learning data is very insufficient as the rolling forecast is applied. Thus, as shown in Fig. 4, the learning data periods employed in these models were gradually increased before the prediction point.

The parameters that are to be determined to the SSL model are k and μ and these parameters represent the number of neighbor node presented in Eq. (1) and the loss-smoothness tradeoff presented in Eq. (2), respectively. Also, the parameter values used in this experiment were determined as an optimal combination for the validation set presented in the range of $\{k, \mu\} \in \{2,3,4,5\} \times \{0.01,0.1,0.3,0.5,0.7,1,10,100\}$. In addition, the optimal value of the hidden node in ANN was determined in the range of {3~50} [5] and the parameters of kernel width (gamma) and misclassification tradeoff (C) in SVM were determined as an optimal combination of the values in the range of $\{gamma, C\} \in \{0.01,0.1,0.3,0.5,0.7,1,10,100\} \times \{0.01,0.1,0.3,0.5,0.7,1,10,100\}$[21].

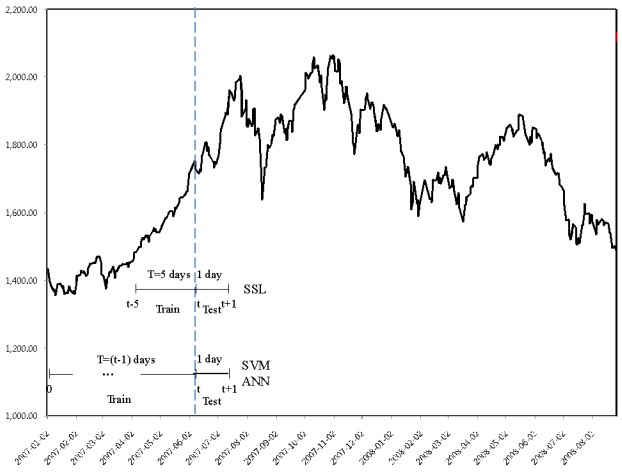


Fig. 4 Experimental setting

V. Results

To measure the prediction performance, the area under the curve (AUC), which is defined as the area under the receiver operating characteristic (ROC) curve [22-23] is used. The ROC curve plots true positive rate as a function of false positive rate for differing classification thresholds as shown in Fig. 5. The AUC measures the overall quality of the model for all possible values of threshold rather than the quality at a single value of threshold. The closer the curve follows the left-hand border and then the top-border of the ROC space, the larger value of AUC the model produces; i.e., the more accurate the model is.

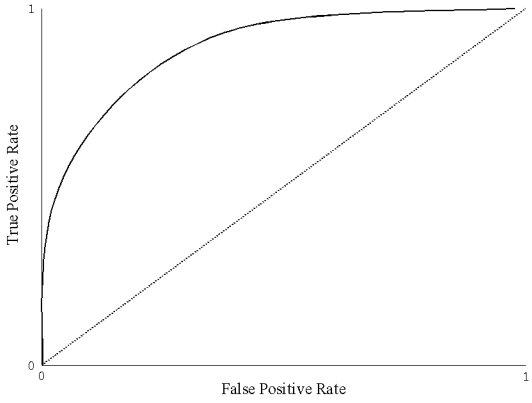


Fig.5 ROC curve

Fig. 6 shows the graph of the values of AUC for the three models used in the test period. Points presented in the graph represent the average section values of AUC in which a section has 10 time points. The average AUC values in SVM and ANN for the total 30 sections were 0.58(±0.08) and 0.51(±0.01), respective, but the value in SSL was 0.72(±0.05). Although ANN represented low volatility based on the standard deviation of 0.01, the AUC value was small compared to other models. In the case of SVM, although it showed partially higher AUC values than SSL, it represented a very high deviation in its accuracy. Whereas, SSL showed stable and high accuracy in most sections compared to that of ANN and SVM. A t-test was applied to verify the significance that SSL represents better performance statistically than that of SVM and ANN. As a result, the difference in the performance between them showed statistical significance as shown in the upper right box in Fig. 6.

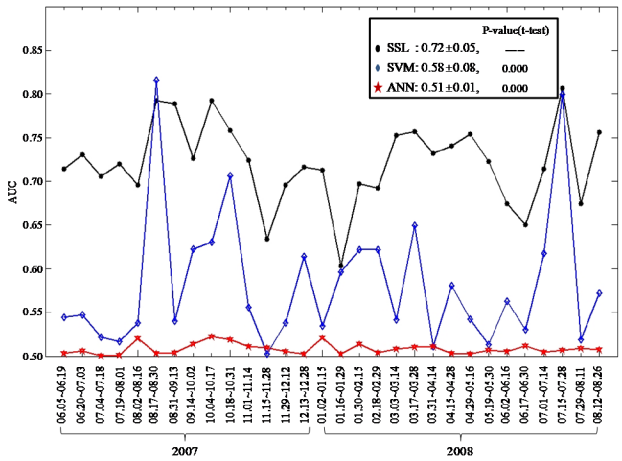


Fig. 6 AUCs comparisons with different methods

VI. Conclusions

In this study, a stock prediction method using time series data to SSL was proposed. The proposed method has the advantage that does not predict stock prices by considering the time series characteristics of the stock price in businesses like the conventional models but makes possible to predict the stock price using a network based on the fluctuation in other companies' stock prices and the economic index that affect the change in stock prices. Regarding the technical issue in the proposed method, the method used SSL and that leads to improve its predictability by including not only the influences on input variables and target variables but also the interrelation between input variables. Based on the combination of these advantages, it was possible to obtain the values of AUC as 0.72. Furthermore, the method proposed in this study can apply for predicting the fluctuation in stock prices for various stock items. Therefore, it is possible to expect profits and stabilities in investments as the results obtained in this study are combined with a portfolio optimization method.

Appendix

Table 1 200 listed stock in KOSPI.

Foods & Beverages	Samyang Corporation, Hite Brew, Doosan Corporation, CJ Corp, Daehan Flour Mills Co, Daesang Corporation, Orion Corporation, Lotte Samkang Co, Namyang Dairy Product Co, Samyang Genex, Nong Shim Co, Lotte Confectionery Co, Bing-grae Co, Lotte Chilsung Beverage Co, Ottogi, Crown Confectionary, Dongwon F&B
Textile & Apparel	Kyungbang Co, FnC Kolon Corp, Nasan, Handsome, The Basic House, LG Fashion Corporation
Paper & Wood	Hankuk paper Mfg, Hansol Paper Co, Seha, Moorim paper.
Chemicals	Woongjin Chemical, Hankook Tire Co, Hanwha Co, Cheil Industries Inc, Kokon, Nexen Tire Co, KCC, Tae Kwang Industrial Co, Samsung Fine Chemicals Co, Hyosung SK Chemicals Co Capro Korea Petro Chemical Aekyung Petrochemical Youl Chon Chemical Hanwha Chemical OCI S-Oil Honam Petrochemical Korea Kumho Pertrochemical SKC UNI D KPX Chemical, KPX Chemical Namhae Chemical, LG Household & Health Care, LG Chem, KP Chemical Corporation, Huchems Fine Chemical Corporation, Kumho Tires, Amore Pacific, Foosung
Medical Supplies	Yuhan Corporation, Ildong Pharmaceutical Co, Dong A Pharmaceutical Co, JW Pharmaceutical, Chong Kun Dang Pharmaceutical, Bukwang Pharm, Ilsung Pharmaceuticals Co, Yungjin Pharm, Dong Wha Pharm, Green Cross, Il-Yang Pharm, Hanmi Pharmaceutical, Kwang Dong Pharm, LG Life Sciences Ltd, Daewoong Pharm
Non-metalic Minerals	Chosun Refractories Co, Dongyang Mechatronics, Hanglas, Asia Cement Co, Hanil Cement Co, Ssangyong Cement Industrial Co, Sung Shin Cement Co, Samkwang Glass Ind Co, Hyundai Cement Co, Hankuk Glass Industries
Iron & Metals	Young Poong Co, Dong Kuk Steel Mill Co, SeAh Be steel Co, Kisco, Kiswire, SeAh Steel, Union Steel, Hyundai-Steel Co, BNG Steel Co, Posco, Poongsan, Korea Zinc, Hyundai Hysco, Dongbu Steel, Daehan Steel
Machinery	Dongbu Hannong Co, KC Cottrell, Shinsung ENG, DKME, Hyundai Elevator, Hankuk carbon, Halla Climate Control co, Doosan Heavy Industries & Construction, Doosaninfracore, Hanmi Semiconductor, STX Engine, Sewon Cellontech, S&TC
Electrical & Electronic Equipment	Hynix Semiconductor Inc, Kumho Electric, Taihan, Daeduck GDS Co, Hansol Lcd Inc, Samyoung Electronics Co, Samsung Electronics, LS Industrial System, Samsung SDI, Daeduck Electronics, Korea Technology Industry, Samsung Engineering, LS, Celrun, Dongwon Systems, Iljin Electric, Korea electric terminal co, Sindoh, LG Display, Hyundai Autonet, LG Electronics
Medicalprecisio	Samsung Techwin, K.C. Tech
Transport Equipment	Hyundai Motors, KIA Motors, Hanjin Heavy Industries, S&T Dynamics, Ssangyong Motor, Hyundai Heavy Industries, Samsung Heavy Industries, Hyundai Mipo Dockyard, Myungsung, Hyundai Mobis, Dongyang Mechatronics, Daewoo Shipbuilding & Marine Engineering, S&T Daewoo STX Offshore & Shipbuilding
Other manufacturing	Fursys, KT&G
Electrical & gas	Kepco, Kogas
Construction	Daelim Industrial Co, Hyundai Engineering & Construction Co, Kumho Industrial Co, GS Engineering & Construction, Hyundai Development, Daewoo E&C
Distribution & Service	Samsung C&T, LG International, SK Networks Co, Amorepacific, LG, SK, Shinsegae Co, STX, S1, Dae Kyo, Coway, Lotte Shopping, Samsung Engineering, Cheil Worldwide Inc, SBS, Kangwon Land, NCsoft, Daewoo International, Hyundai Department Store, GS Holdings
Transport & Storage	Hanjin Shipping Co, Korean Air Lines, Hyundai Merchant Marine
Communication	SK Telecom, KT, KTF
Finance	Samsung Fire & Marine Insurance, Hyundai Securities, Korean Reinsurance, Daegu Bank, Busan Bank, Woori Investment & Securities, Daewoo Securities, Samsung Securities, Industrial Bank of Korea, Mirae Asset Securities, Woori Finance Group, Shinhan Financial Group, Korea Investment Holdings, Hana Financial Group , KB

ACKNOWLEDGMENT

The authors would like to gratefully acknowledge support from Post Brain Korea 21 and the research grant from National Research Foundation of Korean government (KRF-2010-0007804/2012-0000994).

REFERENCES

[1] C.-J. Huang, D.-X. Yang, and Y.-T. Chuang, "Application of wrapper approach and composite classifier to the stock trend prediction," *Expert Systems with Applications,* vol. 34, pp. 2870-2878, 2008.

[2] H.-C. Liu, Y.-H. Lee, and M.-C. Lee, "Forecasting China Stock Markets Volatility via GARCH Models Under Skewed-GED Distribution," *Journal of Money, Investment and Banking,* pp. 5-14, 2009.

[3] H. Amilon, "GARCH estimation and discrete stock prices: an application to low-priced Australian stocks " *Economics Letters,* vol. 81, pp. 215-222, 2003.

[4] N.-F. Chen, R. Roll, and S. A. Ross, "Economic Forces and the Stock Market," *Journal of Business,* vol. 59, pp. 383-403, 1986.

[5] K.-j. Kim, "Financial time series forecasting using support vector machines," *Neurocomputing,* vol. 55, pp. 307-319, 2003.

[6] K. Park, T. Hou, and H. Shin, "Oil Price Forecasting Based on Machine Learning Techniques," *Journal of the Korean Institute of Industrial Engineers,* vol. 37, pp. 64-73, 2011.

[7] W. Huang, Y. Nakamori, and S.-Y. Wang, "Forecasting stock market movement direction with support vector machine," *Computers & Operations Research,* vol. 32, pp. 2513-2522, 2005.

[8] Q. Cao, K. B. Leggio, and M. J. Schniederjans, "A comparison between Fama and French's model and artificial neural networks in predicting the Chinese stock market," *Computers & Operations Research,* vol. 32, pp. 2499-2512, 2005.

[9] A.-S. Chen, M. T. Leung, and H. Daouk, "Application of neural networks to an emerging financial market: forecasting and trading the Taiwan Stock Index," *Computers & Operations Research,* vol. 30, pp. 901-923, 2003.

[10] F. E. H. Tay and L. Cao, "Application of support vector machines in financial time series forecasting " *The International Journal of Management Science,* vol. 29, pp. 309-317, 2001.

[11] A. KANAS, "Non-linear Forecasts of Stock Returns," *Journal of Forecasting,* vol. 22, pp. 299-315, 2003.

[12] B. Yang, L. X. Li, and J. Xu, "An early warning system for loan risk assessment using artificial neural networks " *Knowledge-Based Systems,* vol. 14, pp. 303-306, 2001.

[13] S. Bekiros and D. Georgoutsos, "Direction-of-Change Forecasting using a Volatility- Based Recurrent Neural Network," *Journal of Forecasting,* vol. 27, pp. 407-417, 2008.

[14] P. M. Tsang, P. Kwok, S. O. Choy, R. Kwan, S. C. Ng, J. Mak, J. Tsang, K. Koong, and T.-L. Wong, "Design and implementation of NN5 for Hong Kong stock price forecasting," *Engineering Applications of Artificial Intelligence*, vol. 20, pp. 453-461, 2007.

[15] X. Zhu, "Semi-Supervised Learning with Graphs," *Ph.D. dissertation, Carnegie Mellon University,* 2005.

[16] H. Shin, N. J. Hill, A. M. Lisewski, and J.-S. Park, "Graph sharpening," *Expert Systems with Applications,* vol. 37, pp. 7870-7879, 2010.

[17] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Sch¨olkopf, "Learning with local and global consistency " *Advances in Neural Information Processing Systems* vol. 16, pp. 321-328, 2004.

[18] H. Shin, A. M. Lisewski, and O. Lichtarge, "Graph sharpening plus graph integration: a synergy that improves protein functional classification," *Bioinformatics,* vol. 23, pp. 3217-3224, 2007.

[19] K.-j. Kim, "Artificial neural networks with evolutionary instance selection for financial forecasting," *Expert Systems with Applications,* vol. 30, pp. 519-526, 2006.

[20] M. O'Connor, W. Remus, and K. Griggs, "Does updating judgmental forecasts improve forecast accuracy?," *International Journal of Forecasting,* vol. 16, pp. 101-109, 2000.

[21] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery,* vol. 2, pp. 121-167, 1998.

[22] J. A. Hanley and B. J. McNeil, "The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve," *Radiology,* vol. 143, pp. 29-36, 1982.

[23] M. Gribskov and N. L. Robinson, "Use of receiver operating characteristic (ROC) analysis to evaluate sequence matching " *Computers & Chemistry,* vol. 20, pp. 25-33, 1996.

Finding Interesting Classification Rules: An Application from Education

Anthony Scime **Summer M. Reiner**
Department of Computer Science Department of Counselor Education
The College at Brockport, State University of New York, New York, USA

Abstract - *Classification trees may contain a large number of branches/rules. Some rules are more interesting than others because they (1) perform better than guessing at predicting the class attribute's value while they also apply to a large percentage of the dataset, or (2) contain critical attributes; those attributes important to the problem under investigation or to the domain. Using a dataset from one suburban American high school, C4.5 classification analysis and rule selection revealed the characteristics of the school's biggest disciplinary problem, and the day of the week that students were engaged in each day's biggest disciplinary problem. Additionally the characteristics of students involved in the most serious incidences and the days of those incidences were identified.*

Keywords: Classification, Critical Attributes, Data Mining, Interestingness, Rule Selection

1. Introduction

The analysis of data, through data mining methods, reveals interesting patterns, confirms and probes previously known relationships, and detects previously unknown relationships in data [1]. Classification not only predicts the results of a future event, but also can provide knowledge about the structure and interrelationships among the data. Revealing interrelationships can lead to a better understanding of the data and the domain from which the data is obtained [2].

Classification analysis, constructs a decision tree that provides a path to a predetermined class attribute. The tree's branches are converted into classification rules.

Association mining, which evaluates data for relationships among attributes in the dataset [3], often creates a large number of rules. Significant research has been done in association mining rule selection and reduction [4, 5, 6, 7, 8, 9, 10], to find interesting rules.

Classification trees also can be very large, that is, contain many branches or rules. The dataset's attributes may each contain a large number of values resulting in trees with many branches, in which case the percentage of applicable records for each rule is rather small. Research has been done to reduce the size of the tree, and hence the number of rules by data dimensionality reduction [11, 12, 13]. But, it is not always possible to eliminate attributes. There may not be a sufficient set of attributes, or the domain or problem under investigation may require that critical attributes be part of the classification tree [12, 13, 14, 15].

The number of rules can be reduced by pruning the tree using confidence level, but this effects the tree's accuracy. A classification tree's selection and value is determined by its accuracy in both the creation and evaluation of the tree [2]. That is, the dataset is divided into training and testing sets and the classification algorithm executed at different confidence levels. Each tree constructed with the training dataset is re-evaluated with the test dataset. One of the trees is selected as most representative of the data. This selection is based on the tree's accuracy. When the accuracy for the training dataset most closely matches the accuracy for the test dataset that tree is selected for further study [2].

2. Interesting Classification Rules

However, the tree's overall accuracy is not the same as an individual rule's accuracy. It is the individual rules which need to be applied in the domain to predict the outcome of the class attribute and characterize segments of the domain [16, 17]. As in association mining, some rules are more interesting than others. Interestingness can be measured by objective, subjective, or semantics-based means. Objective interestingness is measured by statistical techniques, which do not consider the specifics of the domain or the problem being considered. Subjective techniques incorporate domain background knowledge, and semantics-based measures consider the goals of the data mining project [18].

Combining a statistical analysis of the dataset with domain or semantic knowledge can lead to interesting rules that are actionable in the domain. The specific goals of the current problem can cause some rules to be interesting and actionable, even though they may not be statistically noteworthy. Interestingness then is based on the predictive ability of the rule, the scope of the rule in addressing the data, or the specific problem under investigation.

The accuracy of a rule is the statistic of the percentage of training records that satisfy the rule compared to the records that meet the rule premise. That is, accuracy is a percentage of those records that are correctly classified by that rule. If a rule is more accurate in predicting the class attribute value than predicting the value without the rule (guessing), it is a candidate interesting rule.

Another objective consideration for rule interestingness is the number of records to which it applies. A rule which applies to a few records maybe 100% accurate but not interesting; whereas a rule that is less accurate while applying to a large percentage of the data's records maybe more interesting. These large percentage record rules are candidate

interesting rules.

Rule accuracy may range from 0-100% and the number of records to which a rule applies ranges from one record to all the records. A method needs to be developed to select interesting rules from the candidate interesting rules. One method combining the accuracy and the number of rules is to select the rules using the standard deviations of the number of correct records per rule and rule accuracy. Consider a rule interesting if its number of records meeting the rule is at least one standard deviation above the average number of records meeting rules (Figure 1) and whose accuracy is no more than one standard deviation below the average rule accuracy (Figure 2).

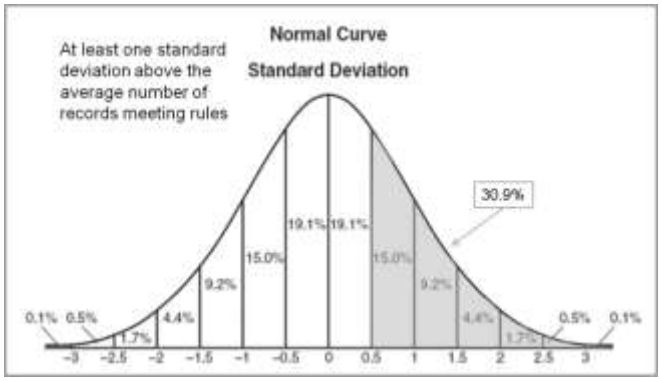


Figure 1. Record Count Measure

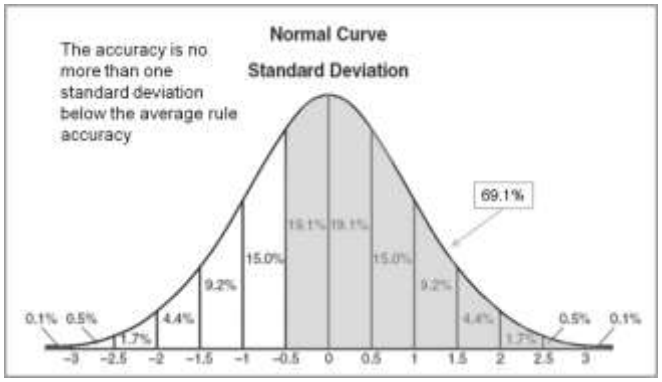


Figure 2. Rule Accuracy Measure

From the domain perspective it may not be possible to take action on all the candidate interesting rules simply based on resource availability. The set of candidate interesting rules can be reduced to only those that satisfy domain constraints. The combination of the objective measures and the subjective domain constraints presents an actionable set of interesting rules.

Additional interesting rules (semantically-based) are those that have attributes critical to the problem under investigation. These critical attributes define the problem but are not necessarily the goal of the classification. Class attributes are critical attributes but in a domain/problem other attributes also may be critical. Critical attributes and their values characterize the problem under investigation, while the remaining attributes characterize the data. Furthermore, some values of the critical attributes may be more significant than other values. Even though the accuracy of rules with these critical attribute value pairs may not be high, the characterization the rules provide may be helpful in achieving the overall data mining purpose.

3. An Example from School Discipline

As in most industries, educators have moved to a model of efficiency and are asked to measure their productivity. Educators are asked to use data to identify systemic issues in schools, which can result in the development of systemic interventions aimed at mitigating identified issues. One such issue is student discipline. School administrators and school counselors spend a significant amount of time responding to student discipline issues [19]. In fact, in a time and task analysis, funded by the Wallace Foundation, School Principals and Assistant Principals reported spending 70%-100% of their time attending to student discipline issues [19]. Predicting student discipline issues, could lead to a reduction of administrators' and counselors' time investment in having to react to such issues. By identifying the student attributes associated with particular discipline issues, one could predict groups of students at potential for receiving specific disciplinary referrals. School administrators and school counselors could then develop targeted prevention and intervention programming aimed at systemically reducing the problem behavior.

By identifying potential issues, counselors develop targeted programming (e.g., classroom guidance, group counseling) to prevent student issues (e.g., absences, use of electronic devices). They use data originally collected to report disciplinary incidences to inform their counseling program [20]. The specific question to be addressed by the data is – who are the students involved in what disciplinary problems on each day of the week.

A dataset was constructed from a suburban American high school's disciplinary data. This dataset contains the 35,272 disciplinary problems occurring in the 2008-2009 school year by students in grades 8-11. For each incident, the grade, gender, ethnicity, primary language, and the special education (IEP), disability (504 plan), English proficiency (LEP), and academic intervention service (AIS) enrollment of each student was recorded, as well as the description (Discipline Description) and the day of the week of the problem (Day of Incident).

All the attributes' values are discrete values. Grade varied from Grade 8 to Grade 11. Gender is male or female, Ethnicity was identified as Black or African American, White, Asian, Hispanic or Latino, and American Indian or Alaskan Native. Primary languages of the students are English, Ukrainian, Croatian, and Greek. In the data there are 45 different discipline descriptions. Finally, all seven days of the week exist in the data. An IEP-LEP-504-AIS attribute was constructed as a combination of the programs in which the student is enrolled. There are 9 combinations of the programs in the dataset.

A simple count of the discipline descriptions finds that the three most common offenses are: missed or skipped class, insubordination, and use of electronics devices. The most

common 18 offenses are fairly well distributed and collectively account for over 45% of the disciplinary issues. Only these 18 offenses occur more than 1% of the time and of those only two are Violence and Disruptive Incidents Reporting (VADIR) offenses - Minor altercations (Assaults) at 1.4% and Other Disruptive Incidents at 1.3%. It should be noted that considering the total size of the student body, this school does not have a large discipline problem.

As is customary in classification, the data was randomly split (approximately 2/3rd and 1/3rd) into two datasets. The training dataset contains 23,516 records and the testing dataset the remaining 11,756 records. The representation of each discipline description is in proportion to the total dataset in both the training and testing sets.

In terms of the days of the week, incidences are spread fairly evenly across the school days at between 17.0% and 22.7%. Again the training and testing sets are proportional representatives of the total dataset.

4. The Disciplinary Classification Tree

The Day of Incident attribute was selected as the class attribute. This results in a tree that defines the day of the week in terms of the disciplinary incidents, a critical attribute, and the characteristics of the offenders on those days. The C4.5 classification algorithm as implemented by the WEKA data mining tool [21] was executed twenty times on the training and testing datasets. In each execution the confidence level was varied from 0.05 to 1.00 by increments of 0.05. The error rate for classification ranged from 42.0% to 42.4% for the training set and 43.6% to 43.8% for the testing set. With a confidence of 0.10 the error rate difference between the training and testing set was the least therefore this tree was selected as the best representative for analysis (Figure 3).

Figure 3. Tree Confidence vs. Percent Error

Overall the selected tree's 1,183 rules predict with 56.4% accuracy the day an incidence will happen and the characteristics of the offender, which is more than twice as accurate as guessing the day of an incident. Given a random incident without the data and assuming a 5-day week there is a 20% chance of correctly guessing the day of the incident. Considering the data, there is a 22.7% chance it happens on a Tuesday. Tuesday is the most common day for incidences, and therefore the best guess, 22.7% of incidences happen on Tuesday. The classification tree can predict the day of the incidence with 56.4% accuracy. However, individual rules may be more or less accurate.

5. Analysis of Results

The tree itself needs to be analyzed. With 1,183 rules some criteria needs to be used to select the interesting rules beyond being more than 22.7% accurate; one such method is to find those rules that apply to the most discipline records. The students missing or skipping class is 22.9% of the incidences in the data and of the 5,393 missed or skipped classes in the training set, most (24.3%) happen on Tuesdays. Concentrating on the 50 missed or skipped class rules, the tree also tells "who is missing class" on each day of the week, and the likelihood that students with those characteristics, will miss or skip class. Each of these 50 rules specifies a description of a student, the day they miss or skip class and an accuracy rate that the rule is correct. To illustrate consider the rule:

IF Discipline Description = Missed or skipped class
AND IEP-LEP-504-AIS = ---AIS
AND Ethnicity = White
AND Gender = Male
AND Grade = Grade 10
THEN Day of Incident = Thursday (1)

This is the missed or skipped class rule that has the most correctly classified incidences (225). It is correct 43% of the time. The rule states that white, male, 10th graders with AIS (and just AIS) are most likely to miss class on Thursdays. These students constitute the largest group (10.1%) of all the missed or skipped classes and 6.5% of all incidents on Thursdays. With a 43% accuracy if you were to accuse one of these students of missing class on Thursday, you would be correct only 43% of the time. But, this student-problem is the greatest student-problem on Thursdays.

Selection of interesting rules can be accomplished by evaluating the rules that apply to the largest number of students and have an acceptable accuracy. To select these rules use the standard deviations of the number of correct records and rule accuracy. Consider only those rules which are at least one standard deviation (56.3) above the average (44.5) number of correct records; and whose accuracy is better than one standard deviation (0.27) below the average (0.63) accuracy. Rule (1) is then an interesting rule. The standard deviation analysis also provides an additional three interesting rules concerning missed or skipped class, those with a high number of correctly identified instances and an accuracy of better than 36%:

IF Discipline Description = Missed or skipped class
AND IEP-LEP-504-AIS = ----
AND Grade = Grade 11
AND Ethnicity = White
THEN Day of Incident = Friday (2)

White, 11th grade, Non-IEP-LEP-504-AIS students of both genders miss class on Friday. The success of this rule is 37%. These students comprise 9.4% of all missed classes and 4.62% of all incidences on Fridays. Note that these students are the greatest missing class offenders on Fridays.

IF Discipline Description = Missed or skipped class
AND IEP-LEP-504-AIS = ---AIS
AND Ethnicity = White
AND Gender = Female
THEN Day of Incident = Tuesday (3)

White, female, AIS students of all grades miss class on Tuesdays. This is true 39% of the time. These students comprise 7.9% of all missed classes and 3.1% of all incidences on Tuesdays. The rule also identifies the greatest class missers on Tuesdays.

IF Discipline Description = Missed or skipped class
AND IEP-LEP-504-AIS = ---AIS
AND Ethnicity = Black or African American
AND Grade = Grade 11
AND Gender = Female
THEN Day of Incident = Wednesday (4)

Wednesdays have a different cohort of missing students – Black or African American, AIS, female, 11th graders. This is correct 40% of the time. This group is 5.5% of all missed classes and 2.8% of incidences on Wednesdays. The rule also identifies the greatest class missers on Wednesdays. Table 1 summarizes the rules with the most number of records for each day of the week. Note that the rule for Monday is not an interesting rule. No Monday rule meets the rule accuracy criteria of greater than 36%.

The greatest problem of each day of the school week (the class attribute) is a critical attribute. The rule for each day of the week with the greatest problem that day informs the administration about what to prevent when. Monday is a skipped or missed detention problem day. Of all the problems on Monday the greatest at only 3.8% is skipped or missed detention by White, grade 11, AIS, males with 31% accuracy and comprising 30% of the week's skipped or missed detentions.

IF Discipline Description = Skipped or missed detention
AND Gender = Male
AND Grade = Grade 11
AND IEP-LEP-504-AIS = ---AIS
AND Ethnicity = White
THEN Day of Incident = Monday (5)

Tuesday's biggest problem is the skipped or missing class problem (rule 3).

Wednesday's leaving school without permission is the biggest problem with 3.3% of the day's occurrences. White, male, AIS, 10th graders leave school on Wednesdays and the application of this rule is correct 80% of the time. This group comprises 19.2% of all students that leave school without permission.

IF Discipline Description = Left school without permission
AND Ethnicity = White
AND Grade = Grade 10
AND Gender = Male
AND IEP-LEP-504-AIS = ---AIS
THEN Day of Incident = Wednesday (6)

On Thursday rule (1) concerning skipped or missing class is the biggest problem at 6.5% of the day's problems.

Friday's biggest problem creates 4.62% of Friday problems, again missing or skipping class (rule 2). Table 2 summarizes the rule for each day of the week with the greatest problem.

Serious but infrequent offenses are the VADIR offenses. VARID incidences constitute a small percent of all the discipline problems at this school. Nevertheless, from the

Table 1. Missed or Skipped Class

Ethnicity	Grade	Special Programs	Gender	Day of Incident	Rule Accuracy	Num Records
White	11	AIS	Male	Monday	34%	160
White	All	AIS	Female	Tuesday	39%	176
African American	11	AIS	Female	Wednesday	40%	122
White	10	AIS	Male	Thursday	43%	225
White	11	None	Both	Friday	37%	209

Table 2. Greatest Day of the Week Problem

Discipline Description	Ethnicity	Grade	Special Programs	Gender	Day of Incident	Rule Accuracy
Skipped Or Missed Detention	White	11	AIS	Male	Monday	31%
Skipped Or Missing Class	White	All	AIS	Female	Tuesday	39%
Leaving School Without Permission	White	10	AIS	Male	Wednesday	80%
Skipped Or Missing Class	White	10	AIS	Male	Thursday	43%
Missing Or Skipping Class	White	11	None	Both	Friday	37%

school administration's perspective, these offenses constitute a critical attribute value pair. The only two VADIR offenses comprising more than 1% of the problems are minor altercations (assaults) at 1.4% and other disruptive incidents at 1.3%.

There are 13 rules characterizing assaults. Assaults occur every day of week but most assaults occur on Fridays and Saturdays. On Friday 2.7% of all incidences are assaults and on Saturdays 5.0% of all incidences. For all other days of the week assaults are less than 1% of the problems.

If there is an assault on Saturday it is 100% likely to have involved an IEP and AIS, male, Black or African American in any grade (rule 7). There are only 17 assaults on Saturdays in the training dataset, all of them involving the same types of students. These are 6% of all assaults.

IF Discipline Description = VADIR Minor altercations (Assaults)
AND IEP-LEP-504-AIS = IEP---AIS
AND Gender = Male
AND Ethnicity = Black or African American
THEN Day of Incident = Saturday (7)

Friday assaults are represented in rules 8, 9, 10, and 11.

IF Discipline Description = VADIR Minor altercations (Assaults)
AND IEP-LEP-504-AIS = IEP---AIS
AND Gender = Male
AND Ethnicity = White
AND Grade = Grade 10
THEN Day of Incident = Friday (8)

IF Discipline Description = VADIR Minor altercations (Assaults)
AND IEP-LEP-504-AIS = IEP---
THEN Day of Incident = Friday (9)

IF Discipline Description = VADIR Minor altercations (Assaults)
AND IEP-LEP-504-AIS = ----
AND Grade = Grade 10
THEN Day of Incident = Friday (10)

IF Discipline Description = VADIR Minor altercations (Assaults)
AND IEP-LEP-504-AIS = ---AIS
AND Ethnicity = White
AND Grade = Grade 10
THEN Day of Incident = Friday (11)

Summarizing Friday assaults, these rules have at least one of the following student characteristics: IEP or AIS students, male students, white students, and 10th graders, A student with all these characteristics (IEP and AIS, male, white, and 10th grade) fits the rule (rule 9) that represents 17.6% of all Friday assaults

There are two other rules that characterize more than 10% of the assault cases. One rule (rule 12) identifies female students in both IEP and AIS as involved in assaults on Mondays. That rule is 100% accurate, there are 41 cases in the training set, and these 41 cases are 14.4% of all the assaults. The other rule (rule 13) characterizes students in IEP and AIS that are 8th grade white, males as involved in Wednesday assaults. This rule is 61% accurate and represents 12.0% of all assault cases.

IF Discipline Description = VADIR Minor altercations (Assaults)
AND IEP-LEP-504-AIS = IEP---AIS
AND Gender = Female
THEN Day of Incident = Monday (12)

IF Discipline Description = VADIR Minor altercations (Assaults)
AND IEP-LEP-504-AIS = IEP---AIS
AND Gender = Male
AND Ethnicity = White
AND Grade = Grade 08
THEN Day of Incident = Wednesday (13)

The remaining six rules concerning assaults represent a small number of occurrences. Summarizing all the assault rules, while no day is assault free, Fridays are clearly the worst and Tuesdays and Thursdays the safest. Students in both IEP and AIS are the most prone to be involved in an assault, while sex and ethnicity are day dependent.

There are only six rules involving the 1.3% of all incidences defined as other disruptive VADIR incidents. The nature of these other incidences is not known.

IF Discipline Description = VADIR Other disruptive Incidents
AND Ethnicity = Black or African American
THEN Day of Incident = Wednesday (14)

IF Discipline Description = VADIR Other disruptive Incidents
AND Ethnicity = White
AND Gender = Female
THEN Day of Incident = Friday (15)

IF Discipline Description = VADIR Other disruptive Incidents
AND Ethnicity = White
AND Gender = Male
AND IEP-LEP-504-AIS = IEP---AIS
THEN Day of Incident = Friday (16)

IF Discipline Description = VADIR Other disruptive Incidents
AND Ethnicity = White
AND Gender = Male
AND IEP-LEP-504-AIS = ----
THEN Day of Incident = Thursday (17)

IF Discipline Description = VADIR Other disruptive Incidents
AND Ethnicity = White
AND Gender = Male
AND IEP-LEP-504-AIS = ---AIS
THEN Day of Incident = Friday (18)

IF Discipline Description = VADIR Other disruptive Incidents
AND Ethnicity = Hispanic or Latino
THEN Day of Incident = Tuesday (19)

Summarizing these six rules, 54.7% of other disruptive incidents occur on Fridays and these represent 2.3% of all Friday incidences. These Friday incidences are most often perpetuated by white, male, IEP and AIS students, regardless of grade. This group accounts for 38% of all other disruptive incidents. In terms of high accuracy and a high percentage of occurrence, the most accurate (75.0%) and the second most occurring incidences of this type (24.5%) are Black or African American students getting into VARID trouble on Wednesdays, and this accounts for 1.1% of all Wednesday problems. The next most serious other disruptive incidents occur on Thursday with 20.3% of the other disruptive incidents caused by white, male, AIS students; this is 1.1% of all Thursdays incidents. Overall most VARID incidences (assaults and other disruptive incidents) occur on Friday.

6. Summary

The analysis was done by classification mining to determine a profile of disciple problems – what is the greatest problem, when problems occur, and who are its most common offenders. Additionally, the most serious (VARID) incidences were also analyzed.

The greatest problem overall is missing and skipping class. This was determined by analysis of the raw data. Given a random student missing class, a guess of which day would be evenly likely (20% chance) on any day of the typical 5-day school week. Familiarization with the data would improve the guess to Tuesday, with a 22.9% likelihood of being correct.

The classification tree identifies groups of students that create an offense and the day it is likely to occur. Knowing the characteristics of a student and using the classification tree, the likelihood of determining a student caused the offense on a given day is more accurate.

Using the tree's rules the likelihood of determining a student missed or skipped class increases from a 22.9% correct guess to between a 34% and 43% correct prediction. This prediction also includes a description of the skippers. The rules characterizing the greatest offenders were identified by considering the number of correctly identified offenders and the correctness of the rule.

The greatest problem on each day of the week (the class and a critical attribute) and the discipline description (a critical attribute) were used to find that for three days (Tuesday, Thursday, and Friday) the greatest problem is the same missing or skipping class problem. Monday's problem is skipped or missed detention day by white, AIS, 11th grade males. This is 30% of all skipped or missed detention. Finally, Wednesday presents a slightly new problem, leaving school altogether. Nineteen and two-tenths of those leaving school come from the rule that they are white, male, AIS, 10th graders leaving on Wednesday and the rule is successful 80% of the time.

According to the rules, overall this school has problems getting students to be where they are suppose to be every day of the week and 10th and 11th grade AIS students are the biggest perpetrators.

Discipline description's values identified as VADIR incidences are the most serious. VADIR problems occurring most often are either assaults or a collection of not specifically classified VADIR incidences. Nevertheless, analysis found that most problems occur on Friday and involve IEP and AIS students, male students, and white students.

7. School Administrator's Actions

This study demonstrated how one suburban American high school was able to predict student disciplinary problems. The C4.5 classification analysis revealed the most frequently occurring disciplinary problems and the characteristics of the students involved. Furthermore, the analysis revealed the day of the week that students were engaged in the identified disciplinary problems. School administrators and counselors could then use the results to: (1) identify issues needing further study to determine a problem's root cause; and (2) mitigate problems before they occur through preventative interventions. Such actions of course will, hopefully, decrease the incidences and change the behavior of the students. This results in a new dataset, which would need to be analyzed again.

8. Conclusion

Schools collect a large amount of data every day. The problems that these data represent need to be addressed. Data mining is a technique that provides an understanding and characterization of the problems.

Classification data mining creates a tree that can answer many questions in a domain. Often some of the attributes in a problem are necessary to help define the problem. These attributes include the class attribute and may also include other attributes that without which there is no problem definition. These attributes are the critical attributes.

The remaining attributes in the data characterize the data itself. They define and segment the dataset.

The large number of rules in a classification tree have varying levels of accuracy, and apply to different numbers of records. The interesting rules are those that have an acceptable accuracy and apply to a significant number of records. Here, these rules were determined as meeting two criteria. The accuracy needed to be not more than one standard deviation below the average and the number of records one standard deviation above the average.

Secondly, interesting rules are those that have a critical attribute with specific values. The class attribute is a critical

attribute and the rules concluding with specific classes and having the most applicable records in that class are interesting rules. Other attributes may also be critical. These critical attributes may have values of special interest to the domain or problem. The rules containing these critical attribute value pairs are interesting rules, without regard to record count or accuracy.

The example analysis was done by classification mining to determine a profile of disciple problems – what is the greatest problem (the critical attribute Discipline Description), when problems occur (the class attribute Day of Incident), and who are its most common offenders (the characterization attributes). Finally, critical attribute value pairs for the most serious (VARID) incidences were also analyzed.

References

[1] C. Zhao and L. Luan, "Data mining: Going beyond traditional statistics," *New Directions for Institutional Research*, pp. 7–16, 2006.

[2] K-M. Osei-Bryson, "Evaluation of decision trees: A multicriteria approach," *Computers and Operations Research*, vol. 31, No. 11, pp. 1933–1945, 2004.

[3] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases,"'in *Proc. 1993 ACM SIGMOD International Conference on Management of Data*, Washington, DC, pp.207–216.

[4] W. Li, L. Han, and J. Pei, "CMAR: Accurate and efficient classification based on multiple class-association rules," in *Proc.* 2001 *IEEE International Conference on Data Mining*, San Jose, CA, pp. 369–376.

[5] S. Jaroszewicz and D. A. Simovici, "Interestingness of frequent itemsets using Bayesian Networks as background knowledge," in *Proc. 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Seattle, WA, 2004, pp.178–186.

[6] N, Zhong, Y. Yao,Y. M. Ohshima, and S. Ohsuga, "Interestingness, peculiarity, and multi-database mining," in *Proc. First IEEE International Conference on Data Mining*, San Jose, CA, 2001, pp.566–574.

[7] Y. Zhao, C. Zhang, and S. Zhang, "Discovering interesting association rules by clustering," AI 2004: Advances in Artificial Intelligence, vol. 3335,Heidelberg: Springer, 2005, pp. 1055–1061.

[8] M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A. I. Verkamo, "Finding interesting rules from large sets of discovered association rules." in *Proc. Third International Conference on Information and Knowledge Management*, Gaithersburg, MD, 1994, pp. 401–408).

[9] H. Toivonen, M. Klemettinen, P. Ronkainen, K. Hätönen, and H. Mannila, "Pruning and grouping of discovered association rules," in *Proc. ECML-95 Workshop on Statistics, Machine Learning, and Discovery in Databases*, Heraklion, Crete, 1995, pp. 47–52.

[10] R. Srikant and R. Agrawal, "Mining generalized association rules," in *Proc. 21st VLDB Conference*, Zurich, Switzerland, 1995, pp. 407–419.

[11] X. Fu and L. Wang, L., "Data dimensionality reduction with application to improving classification performance and explaining concepts of data sets,". *International Journal of Business Intelligence and Data Mining*, vol. 1, No. 1, pp. 65–87, 2005.

[12] G. R. Murray, C. Riley, and A. Scime, "A new age solution for an age-old problem: Mining data for likely voters," presented at the 62nd Annual Conference of the American Association of Public Opinion Research, Anaheim, CA, 2007.

[13] A. Scime, and G. R. Murray, "Vote prediction by iterative domain knowledge and attribute elimination," *International Journal of Business Intelligence and Data Mining*, vol. 2, no. 2, pp. 160–176, 2007.

[14] M. Hofmann and B. Tierney, "The involvement of human resources in large scale data mining projects," *Proc. 1st International Symposium on Information and Communication Technologies*, Dublin, Ireland, 2003, pp. 103–109.

[15] S. S. Anand, D. A. Bell, and J. G. Hughes, "The role of domain knowledge in data mining," *Proc. Fourth International Conference on Information and Knowledge Management*, Baltimore, MD, 1995, pp. 37–43.

[16] A. Scime, G. R. Murray, and L. Y. Hunter, "Testing terrorism using iterative expert data mining," *Proc.2009 International Conference on Data Mining*, Las Vegas, NV, July, pp. 565–570.

[17] G. R. Murray and A. Scime, "Microtargeting and electorate segmentation: Data mining the american national election studies," *Journal of Political Marketing*, vol. 9, no. 3, pp. 143–166, 2010.

[18] L. Geng and H. J. Hamilton, "Interestingness measures for data mining: A survey," *ACM Computing Surveys*, vol. 38, no. 3, Art. 9, September 2006.

[19] B. J. Turnbull, M. B. Haslam, E. R. Arcaira, D. L. Riley, B. Sinclair, and S. Coleman. (2009, October). Evaluation of the school administration manager project [Online], Available: http://www.wallacefoundation.org/knowledge-center/school-leadership/effective-principal-leadership/Pages/The-School-Administration-Manager-Project.aspx

[20] D. Finkelstein. (2009, May). A closer look at the principal-counselor relationship: A survey of principals and counselors, Available: www.schoolcounselor.org/files/CloserLook.pdf

[21] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining Practical Machine Learning Tools and Techniques*, San Francisco, CA: Morgan Kaufmann, 2011.

Using Random Probes for Neural Networks Based Features Selection

Hazem Migdady
Department of Computer Science
Southern Illinois University, Carbondale IL

Norman Carver
Department of Computer Science
Southern Illinois University, Carbondale IL

Abstract- *Feed forward artificial Neural Networks with backpropagation learning algorithm are of the efficient classification and pattern recognition tools that are robust to noise and can learn the target function of many learning tasks. Even though it still suffer of the long training time which limits its efficiency especially over online tasks and high dimensional datasets it is still desirable. In the context of improving neural network efficiency, it is useful to apply features selection principles that can reduce the number of neural network inputs which in turn reduces the required training time and enhances its generalization capability to end up with a neural network based classifier that perfectly matches the selected features set with good classification accuracy.*

Keywords: Neural Networks, Features Selection, Classification, Machine Learning, Random Probes.

1. Introduction

FEED forward artificial Neural Networks with the backpropagation learning algorithm are efficient tools in learning tasks that involve classification and pattern recognition. Neural networks are robust to noise and have the ability to converge to the target function to be learned by approximating the values of the desired target functions. [1] In his definition for neural networks, Negnevitsky believes that:

"[Neural Network] *is a model of reasoning based on the human brain. Thus, a neural network is considered as a highly complex, nonlinear, and parallel information processing system*" [2]

Even though multilayer feed forward neural networks with backpropagation learning algorithm are computationally expensive due to long training times, they are still desirable since they can converge over many learning tasks. Thus, many efforts have been done to improve backpropagation neural networks performance and increasing its efficiency, especially its generalization and classification ability.

A critical factor that affects neural networks performance is the number of its inputs. Experiments from previous works have shown that an abundance of a neural network inputs (*i.e.* features) often results in overfitting and poor generalization for the classifier, while if less inputs than necessary are used this limits the efficiency and capability to converge to the target function. Features can be categorized as: 1) relevant features and 2) irrelevant features which may be contained in any dataset.

Relevant features are those affect the underlying structure of the data and provide enough information about the target, while irrelevant features do not. [3]

In reference [4] the authors defined relevant and irrelevant features using the conditional probability. Under the assumption that the feature values are discrete, F is a random variable of features $[F_1, F_2 \dots F_n]$ then a pattern vector $f = [f_1, f_2 \dots f_n]$ is a realization of F. Hence feature F_i is surely irrelevant iff for all subset of features K^{-i} including F^{-i}:

$$P(F_i, Y|K^{-i}) = P(F_i|K^{-i})P(Y|K^{-i})$$

F^{-i} is a subset of F excluding feature f_i, K^{-i} is a subset of F^{-i} and Y is the target which is a random variable taking values y. This definition implies that feature F_i does not affect the value of the target variable Y (*i.e.* Y is independent of F_i), hence F_i is irrelevant to Y.

In this paper, we introduce a novel method that applies features selection concepts and sieves the original candidate features in a dataset to explore its intrinsic dimensionality and to remove irrelevant features. This improves neural network performance by recognizing and keeping relevant features, which enhances the generalization capability of the classifier and saves resources in any future data gathering.

2. Related Work

A number of approaches and techniques have been proposed to overcome the curse of high dimensional datasets (*i.e.* dataset with a large amount of features) that limits the efficiency of a multilayer feed forward neural network. A dataset with a large number of features implies that the neural network receives a large number of inputs, which in turn increases the required training time and computations, especially in fully connected networks.

Features selection methods can be categorized mainly onto two categories: 1) Filters and 2) Wrappers. Filters are techniques that select features without taking into account the optimization of a learning machine topology and its performance (*i.e.* preprocessing, predictor independent step –

model free techniques). On the other hand, wrappers involve the process of predictor optimization as a correlated step to the features selection process. Thus, even though they are computationally more intensive, they have the advantage of avoiding the problem that the selected features subset with filters may not match the selected predictor perfectly, which may result in poor performance of a classifier. In the case of wrappers, a learning machine performance is utilized to evaluate features subsets according to their predictive power. [4]

In their approach, Heuristic for Variable Selection (HVS), Yacoub and Bennani [5] suggested a feed forward neural network based wrapper that tries to reduce the number of input features according to network weights behavior during training process. At each training session, the input feature with the lowest weights will be pruned. Even though HVS is simple and easy in the sense that it does not involve complicated calculus and it has good results in comparison with other methods, it requires a number of training sessions equal to the number of irrelevant and redundant features to be pruned, which might be large. This can limit HVS performance by increasing the required search time, taking into account that the search depends on the classifier performance.

Another method described in [3] contains two phases, a filter phase followed by a wrapper phase. The filter phase sieves features using a genetic algorithms technique. The second phase starts as a wrapper by presenting the selected features from the first phase as inputs to a feed forward neural network, in order to recognize and remove redundant features according to that network's performance. This method removes features by filtering without taking into account the performance of the produced classifier, since the fitness function in the genetic algorithm evaluates features according to cost and inconsistency measures which are not critically related to the classifier performance. Moreover this method consumes a large amount of neural network training even though it is not necessary to retrain the network after each reduction. The authors tried to overcome the problem of training the neural network over the entire set of features by using genetic algorithms, but genetic algorithms also involve a large amount of computation.

It is possible to note that most of the methods which aim to improve neural network performance are based on the weights behavior during the training process of the network. The main limitation for a neural network is its training cost over the entire features in a dataset especially over those with high dimension. Thus, some methods try to reduce the number of features in a separate step before presenting data to the neural network, as in [3]. Even though such an approach saves time, it has risks removing some features that may have a critical effect of improving classifier performance in combination with other features.

Another method that follows the same strategy was proposed in [6]. This method is a filter method that starts by ranking all features using Gram-Schmidt orthogonalization technique. [7] After that, a threshold is ranked and inserted in that list. This threshold acts as a boundary between relevant and irrelevant features. All features that are ranked lower than that threshold will be discarded since they are considered irrelevant features. After that, the selected features are presented to a fully connected feed forward neural network to be trained. In this approach the classifier is not involved in the features selection process which reflects on its performance since the selected features are not necessarily will match the classifier perfectly even though the classifier will converge over them.

An approach uses the sensitivity analysis for features selection was explained in [8]. The main objective of the sensitivity analysis is to find the saliency of each feature individually. Except the feature under consideration all features are assigned the mean of their values while training the neural network over the current feature. This process is repeated for all features. Then a random phantom feature is used to compare the saliency of all features to its saliency, thus each feature with saliency less than the phantom's one will be discarded. This method suffers of the massive computations since it trains the neural network to find the saliency for all features, which implies a number of trainings that equal to the number of features.

In his approach, Zhang suggested an evolutionary combination between neural network and genetic algorithm. This method performs the features selection process and the network optimization simultaneously to produce a neural network based classifier. [9] Actually the main disadvantage here is the large amount of computations to end up with the classifier.

3. Problem Definition

Multilayer feed forward neural networks are robust to noise learning machines that perform classification and pattern recognition tasks. Previous experiments, mentioned in section II, showed that the performance of a neural network over a dataset is affected by that data set features. How many features and which features should be presented to the neural network are two critical factors affecting the performance. Since irrelevant features have negative effect on a neural network's classification and generalization ability, removing such features will increase accuracy and efficiency, saves resources, and critically reduce the required training time. Several approaches have been proposed to sieve and select a set of features that match a neural network based classifier, as mentioned in the previous section. Some of those approaches combine the process of features selection and neural network topology optimization together (wrappers) while some of them do not. Even though such

combination comes at price by increasing the required amount of computations, it produces more efficient and effective classifiers. Some of the methods that try to overcome the problem of using neural networks over high dimensional datasets require a long training time, while most of them need a large amount of computations. The methods which avoid the long training time perform the entire features selection process or, at least part of it, independently of the neural network optimization, which in turn exacerbates the problem of matching features with learning machine.

3.1 BFSW: Binary Features Selection Wrapper

The proposed method is considered as a wrapper method since it involves both the features selection process and neural network optimization process.

BFSW tries to find a fully connected feed forward neural network over the lowest possible number of features in a dataset with good classification accuracy. As figure 1 illustrates, BFSW starts by training the neural network over the entire candidate features in the dataset to produce a decreasing ordered features ranked list. After that, the same network will be trained over *random probes*, to generate a threshold that separates relevant from irrelevant features.

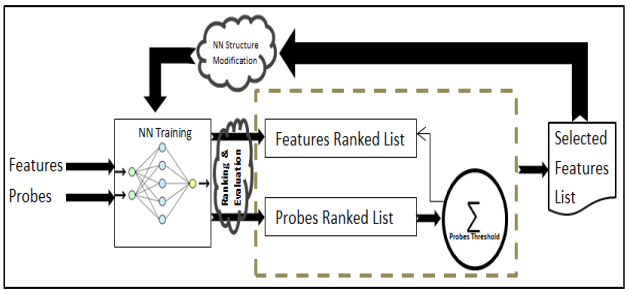


Fig. 1
The General Process of BFSW

3.2 Relevance Index

Although BFSW combines and performs the process of features selection and neural network optimization simultaneously, it avoids the complicated calculus, the long training time, and the massive amount of computations that appear in the methods those have been mentioned earlier. BFSW exploits the weights behavior of the neural network during the training session over the candidate features. Those weights can be understood as indicators of the importance of a specific feature and its contribution to the target output. Note that this is true if all features are normalized into the same range. Thus, after training the neural network over the entire dataset, it is possible to calculate the *relevance index* of each feature. A feature's relevance index S_i is "*relevance quantitative assessment of a candidate feature and the target output*". [10] S_i is calculated according to a feature's final weights which connect it to the output layer through the hidden layer, when that neural network converges, as equation 1 shows [13]:

$$S_i = \sum_{o \in O} \sum_{j \in H} |w_{oj} w_{ji}| \qquad (1)$$

H, *O* denote the hidden and the output layers respectively. While i is a node (*i.e.* feature) in the input layer. The inner term is the product of the weights from input unit i to hidden unit j, and from hidden unit j to output unit o. The sum of the absolute values of those products over all connections –in the network- from unit i to unit o is the relevance index of feature i that shows its contribution and importance to the output.

In BFSW, to calculate the relevance index of all features in a dataset, we need to train the neural network over the entire set of features only once, which saves training time.

After calculating the relevance index of all features, a decreasing ordered ranked list of the features is produced. Features with high relevance indices of that list (*i.e.* highly informative about the target) are considered as relevant features, while those with low relevance indices (*i.e.* poorly informative about the target) are considered as irrelevant. The problem is that there is not a specific boundary that separates relevant features from irrelevant features in that list. Thus, it is necessary to find a suitable threshold that separates those two parts from each other. That's accomplished using random probes.

3.3 Random Probe Threshold

One of the techniques that can be used to calculate a threshold is the random probes technique. This technique was first suggested in [11]. Random probes are irrelevant features could be generated by shuffling the candidate features vectors in the matrix of training data. [12] This shuffling is done by keeping the targets as they are and randomly swapping candidate features vectors. This results in irrelevant features vectors for the targets. This process keeps the features patterns as they are while producing inconsistency with the target values in comparison with the original dataset. After generating the probes, they will be presented to the same neural network that was used in ranking the candidate features. When the classifier converges over the probes, the relevance indices for all probes will be calculated using equation 1 exactly as what was done then with the candidate features.

The random probe threshold will be generated from the relevance indices of the probes. The random probe threshold is the average of all random probes relevance indices, and it is calculated by using equation 2:

$$P_t = \left(\left(\sum_{i \in I} p_i \right) / n \right) \qquad (2)$$

P_t is the probe threshold, i is an input node in the input layer I, while p_i is the relevance index of random probe i which is equivalent to S_i in equation 1 after applying that equation over the final weights of the random probes instead of the candidate features. What equation 2 does, is calculating the average of the probes relevance indices over the total number of the candidate input features which is denoted as n, and thus producing the random probe threshold P_t.

Since a neural network is robust to noise and adaptive in nature, it will converge over the probes even though they are irrelevant features. Hence the random probe threshold will appear somewhere in the candidate features ranked list. This probe is used as a boundary that separates relevant features of the decreasing ordered ranked list from irrelevant features. Thus all features with relevance indices greater than the probe threshold will be kept, while those with lower relevance indices will be discarded. As illustrated in figure 1, after producing the set of the selected features (*i.e.* relevant features) the network structure will be optimized and retrained once again over the selected features. The performance of the new classifier over the selected features will be compared to the performance of the previous one. This process is repeated till the stopping criterion is satisfied. The stopping criterion of this method is based on the classifier's performance over unobserved examples, which is assessed after each training session, directly after classifier structure optimization and the training of the new classifier. Thus, as long as the model performance increases (*i.e.* gets better) in comparison with the performance of the model from the previous training, the process of network training and features reduction goes on. When the classifier performance decreases, the process terminates and the NN classifier with the best performance is chosen.

BFSW takes the structure optimization of a neural network into consideration during the process. At each training session the number of the units in the hidden layer should be twice the number of input units in the input layer, while the number of input units is always equivalent to the number of the selected features at that training session (*i.e.* each input unit receives only one input feature). Figure 2 illustrates the general representation of the hypotheses in the hypothesis space which contains every possible neural network based classifier.

The proposed method is not as straightforward as it seems. Actually an important question arises, which is: *what if the new classifier (after features selection and structure optimization) does not have a better performance than that of the previous one?* The answer to this question has a critical effect on the overall performance of this method. Till now, such a case forms the stopping criterion of the process as mentioned earlier. However we believe that it is not wise to terminate the process at that point, because such an approach does not guarantee that the classifier from the previous training session is the best possible one that could be produced by BFSW, better classifiers could be produced. Thus, we will use a features selection heuristic to circumvent this issue which is: "*individually irrelevant features may become relevant when used in combination*". [4]

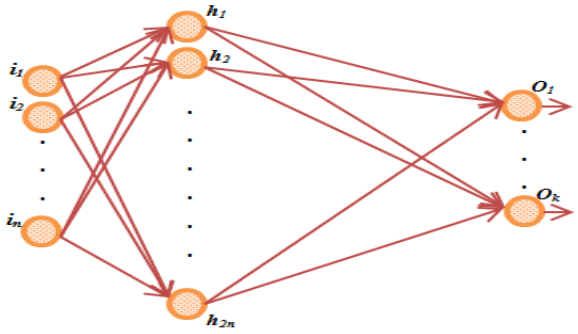


Fig. 2
Neural Network Based Classifier

According to such assumption the proposed method should not terminate just because the new classifier has not better performance than that of the previous one. What we need to do is to control the stopping criterion to assure that the produced classifier is the best possible one that could be produced by BFSW, taking into account that this method does not perform a comprehensive search through the hypothesis space. It is possible to achieve this by making the method more dynamic and directing the search process to reach better solutions.

BFSW tries to find good classifier by applying the "Best First" search technique which can "*back-track to more promising previous subset of features and continue the search from there when the path being explored begins to look less promising*". [15] Every hypothesis in the hypothesis space, as it is illustrated in figure 2, is a fully connected feed forward neural network in which the number of the hidden units is always twice the number of input units which are equivalent to the number of candidate features at the current training session. The hypothesis space is relying on a very useful ordering structure, which is: a Specific-to-General ordering of hypotheses. The most specific hypothesis (*i.e.* the null hypothesis) is that receives the entire set of candidate features as inputs.

The search process starts by enumerating the most specific hypothesis. Then and according to its performance the search direction is directed to those hypotheses which may have better classification accuracy.

Assume that the *n*-classifier (*i.e.* the produced classifier at training session *n*) has a better performance than that for *n+1*-classifier. At that point the current stopping criterion would be satisfied, which implies that the process should terminate and the best classifier is the n^{th} classifier. Since the discarded features at session *n* caused the performance of the

produced classifiers to decrease, applying the "Best First" search technique is considered as a reasonable choice because it has the ability to back-track to explore more promising subsets. Since the classifier performance is decreased after discarding a specific set of features, the search process should be applied as described earlier to sieve features in the discarded features set at training session *n* and manipulate it as all features sets have been manipulated in the previous training sessions. Thus, the discarded features set at training session (*n*) will be divided into two parts according to its corresponding probe threshold that is generated by its corresponding discarded probes set (taking into account that the random probe threshold at any training session divides the probes list into two parts as well as it does with the candidate features set). Hence, the upper part of that list will be chosen and appended to the originally selected one to form together the extended selected features set. If the classifier performance over the extended selected features set outperforms that of the previous classifier (*i.e.* classifier produced at training session *n*), then the process proceeds over the selected features set as explained before. Otherwise the extension and the search process should be applied over the discarded features set once again. This process goes on while the discarded features set contains more than one feature and the performance of the produced classifier is still decreasing, at that situation the process terminates and the classifier with the highest performance is eventually chosen among all produced classifiers.

Even though the search strategy here does not enumerate every possible hypothesis in the hypothesis space either directly or indirectly, it is efficient since it avoids the exhaustive search which consumes long time, and eventually produces an efficient classifier.

4. Implementation and Results

BFSW was implemented and some preliminary experiments ran in order to assess the effectiveness of this combination. The classifier is a multilayer feed forward neural network in which the number of the units in the input layer is equivalent to the number of the candidate features at the current training session, while the units' number in the hidden layer is always twice the number of units in the input layer. The classifier is trained over the candidate features, the random probes, and the selected features using backpropagation learning algorithm and the same weights initial values those were used at the first training session. Actually, the neural network is trained over the random probes only once (at the first training session) and the produced relevance indices are used during the rest of the training sessions without any need to retrain the network over them once again.

The training and testing processes were performed over the SPECT Heart and Breast Cancer datasets, real problems. [14] Figure 3 shows the results of the BFSW implementation over those datasets.

Figure 3 illustrates that the entire candidate features are 31 for the Breast Cancer and 22 for the SPECT Heart. In both datasets BFSW was able to recognize irrelevant and redundant features and make a decision to rule them out while keeping the most informative (*i.e.* relevant) features by using the random probe threshold and the classifier weights as relevance indices. It is possible to note that the number of features was roughly reduced from 22 to 9 and 31 to 7, for SPECT Heart and Breast Cancer respectively while the classifier overall classification accuracy was improved. The classification accuracy increased after features selection process.

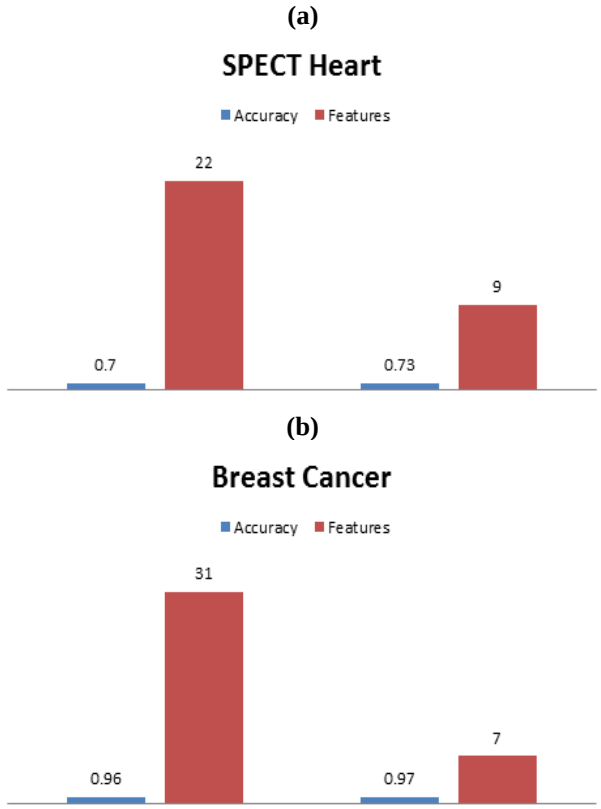


Fig 3:
Implementation Results over SPECT Heart and Breast Cancer Datasets

For SPECT Heart the classification accuracy improved to 0.73 over 9 features instead of 0.7 over 22 features, and it was also improved for the Breast Cancer to be 0.97 over 7 features instead of 0.96 over a set of 31 features. Even though the classification accuracy was slightly increased, it is better to have 0.97 or 0.73 of classification accuracy over a small set of features than having lower or even the same classification accuracy over the entire set of features. During the implementation we noticed that BFSW consumed only one training session to converge to global maxima over the

SPECT Heart, while for Breast Cancer it consumed two training sessions. This variation appears because each data set has its own characteristics, patterns, and correlations which affect the performance of BFSW and make it vary over different datasets. Moreover BFSW is considered as a random technique, since it is based on neural networks which are initialized randomly and the random probes which are generated randomly. Such factors interpret the variation of the BFSW over different datasets.

An important advantage of this method is the simplification of random probes usage. As said before, random probes approach was first suggested in [11]. The major aim of this approach is to reduce the number of the candidate features independently of the learning machine, thus it is considered as a filter approach. In order to make sure that the selected features, after applying random probes, are sufficient to perform the learning task, the decision of discarding features is supported by a statistical test. More details about traditional usage of random probes are available in [4]. BFSW used the random probes in a different and simple way since it makes the decision of evaluating and discarding features directly related to the learning machine performance which in turn helps to avoid the required statistical tests. Random probes are normally used with the Gram-Schmidt technique which was first suggested in [7]. BFSW replaced Gram-Schmidt with the neural network and used its weights as relevance indices for the candidate features. This way makes the relevance indices more informative about the effect and the importance of each candidate feature.

5. Conclusion and Future Work

To sum up, the proposed method is a wrapper method that aims to find sufficient features subset that is convenient to match a neural network based classifier by searching a hypothesis space which has a naturally occurring Specific-to-General ordering structure. The search process is implemented under the following assumption: "*the solution exists in the hypothesis space*". The major aim of BFSW is to simplify and speed up the search process to reach the required hypothesis in the hypothesis space, by applying an efficient search technique than those used in some of the currently existing methods. This method is a novel one since it uses random probes in a wrapper technique and combines it directly with the learning machine. In BFSW the neural network topology is taken into consideration and it is optimized at each training session. The preliminary results of BFSW are promised results and showed the possibility and the efficiency of combining random probes with neural networks. In the future work, we are to implement BFSW with more challenging datasets and compare its results with the currently existing methods.

6. References

[1] T. Mitchell, *Machine learning*. Singapore: McGRAW-HILL, 1997.

[2] M. Negnevitsky, *Artificial intelligence: a guide to intelligent systems*. Britain: Addison-Wesley, 2005.

[3] H. Yuan, S. Tseng, W. Gangshan, and Z. Fuyan, "A two-phase feature selection method using both filter and wrapper," *IEEE International Conference*, vol. 2, 1999, pp. 132 – 136.

[4] I. Guyon, M. Nikravesh, S. Gunn, and L. A. Zadeh, *Feature extraction – foundations and applications*. Berlin Heidelberg New York: Springer 2006.

[5] M. Yacoub and Y. Bennani, "HVS: a heuristic for variable selection in multilayer artificial neural network classifier," *Artificial Neural Networks in Engineering*, 1997, pp. 527-532.

[6] M. Stricker, F. Vichot, G. Dreyfus, and F. Wolinski, "Two-step feature selection and neural network classification for the trec-8 routing," in *Proc. of the Eight Text Retrieval Conf.* 1999. http://arxiv.org/ftp/cs/papers/0007/0007016.pdf

[7] S. Chen, A. Billings, and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *International journal of control*, vol. 5, 1986, pp. 1873-1896.

[8] M. J. Embrechts, F. Arciniegas, M. Ozdemir, C. M. Breneman, K. Bennett, and L. Lockwood, "Bagging neural network sensitivity analysis for feature reduction for in-silico drug design," in *IEEE International Joint Conf.*, 2001, pp. 2478–2482.

[9] B. Zhang, "A Joint evolutionary method based on neural network for feature selection," in *IEEE Second International Conf. on Intelligent Computation Technology and Automation*, 2009, pp. 7 – 10.

[10] H. Stoppiglia, G. Dreyfus, R. Dubois, and Y. Oussar, "Ranking a random feature for variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, 2003, pp. 1399-1414. http://remidubois.free.fr/publications/118.pdf

[11]H. Stoppiglia, "Methodes statistiques de selection de modeles neuronaux, application financieres et bancaires," PhD. Dissertation, Universite Pierre et Marie Curie, Paris, 1997.

[12] J. Bi, K. P. Bennett, M. Embrechts, C. M. Breneman, and M. Song, "Dimensionality reduction via sparse support vector machines," *Journal of Machine Learning Research*, vol. 3, 2003, pp. 1229-1243. http://homepages.rpi.edu/~bennek/papers/bi03a.pdf

[13] P. Leray, and P. Gallinari, "Feature selection with neural networks," *Behaviormetrika*, vol. 26, 1999, pp.145–166.

[14] "Machine Learning Repository", http://archive.ics.uci.edu/ml/

[15] M. A. Hall. "Correlation-based Feature Selection for Machine Learning." PhD, The University of Waikato, Hamilton, NewZealand, 1999.

Fraudulent Bill-Claim Detection in Health Insurance

Junwoo Lee, Juhyeon Kim, Hyunjung Shin*

Abstract— **Fraudulent and abusive bill claims by medical care providers incur physical and fiscal costs to society. In order to identify them, a variety of indices have developed and evaluated diverse aspects of bill claim pattern. When taking all of indices into account, however, it becomes confusing to find out which index is of more importance than others, and even more difficult if the respective results are significantly discordant. To avoid the ambiguities, we propose a method that efficiently quantifies the degree of anomaly in the respective indices and then integrates them based on Genetic Algorithm. When tested on the Korean Health Insurance Review and Assessment data, the proposed method showed promising result of avg. 0.965 AUC, significantly outperforming the competing models including regression, neural network, and decision tree, etc.**

I. Introduction

Recently, the national medical expenses in South Korea is radically increasing. While the total expenses for medical treatment in 2002 had been 13,800 billion won, in 2007 it increased by 2.5 times and was 32,259 billion won, which shows a great upswing. The reasons for the increase include the development of medical facilities and the increase of elderly population, which are seen in natural aspects. On the other hand, they include negative aspects such as medical fraud. According to the report by NHCAA (National health Care Anti-fraud Association), it was assumed that 3~1-% of the total medical expenses in the United States (60 ~160 billion USD) were lost due to medical fraud. Judging from this result, we can expect that there would be a great scale of loss due to medical fraud in South Korea as well. Therefore, many studies on detection of medical fraud have been conducted to reduce the loss. In fact, however, there are many difficulties in terms of professional or technical aspects regarding a domain. First of all, the medical data requires professional understanding and its volume is tremendous. Because there is limited number of data processing professionals who have knowledge of a domain in South Korea, it is not easy to develop the appropriate system which deals with medical fraud. For the additional difficulty in terms of technical aspect, because the pattern of medical fraud is irregular, the fraud detection model generated from previous data is not suited for new data. For this reason, the previously developed detection system ends on the level of research and is rarely applied in practical setting. Therefore, the unjust or false claims have been detected manually by a few professionals. However, in reality we almost reach the limits in using the conventional method to detect the medical fraud which are getting developed over time while the data is greatly increasing.

The authors would like to gratefully acknowledge support from Post Brain Korea 21 and the research grant from National Research Foundation of Korean government (KRF-2009-0065043/ 2012-0000994).

* Corresponding author: Hyunjung Shin (shin@ajou.ac.kr) is a professor of the department of Industrial & Information Systems Engineering, Ajou University, Suwon, 443-749, Korea

Since the detection of medical fraud is a kind of detection of fraud, it is similar field to the detection of insurance fraud or credit card fraud. While the fraud detection system for insurance or credit card companies has been developed in response to the commercial requests of these companies, the fraud detection system in medical field has been developed based on academic research because medical field does not highlight commercial aspects compared to these companies.

According to the previous studies on medical fraud, the following methods were used: conventional statistical methods, data mining algorithms, and machine learning methods. The most mentioned methods in relevant studies are the Neural Network which exerts excellent effect on complicated data [4, 8] and the Decision Tree which is easily applied in practical setting because it makes interpretation of results easy [2]. The part of the studies conducted abroad was actually applied to the medical fraud system of each country and it showed high achievements compared to the manual labor done by a few professionals in the past. In Utah, the United States, they sort out the claim patterns which are suspected of unjust claims by analyzing data through data mining [9]. In Texas, the United States, they detected 1,400 fraud cases by using fraud detection system and collected 2.2 million USD [1]. The HIC of Australia separated meaningful rare data by applying genetic algorithm and k-NN algorithm [3,5]. It sorts out the claim patterns automatically, which was done manually by professionals [9]. The National Health Insurance (NHI) of Taiwan applies the clinical pathways to detect unknown unjust claims. The clinical pathway is a guideline for medical diagnosis and treatment that is defined by certain disease. Through this clinical pathway, they disclose the actions deviated from normal procedures for medical diagnosis and treatment [11].

The Health Insurance Review & Assessment Service (HIRA) of South Korea makes various claim indices and investigates the hospitals, clinics, dental clinics and oriental medicine clinics that are suspected of abuse or unjust use of medical expenses (hereafter, these institutions are called 'problematic institutions' in this paper). The current detection method has two problems by and large. The first problem is that the detection is not made based on the quantitative value, but made based on the order. If the judgment whether the institution is abnormal is made based on the order, it is impossible to show the difference between the problematic institutions and the rest of institutions by expressing numerically. It is difficult to show the level of severity. Therefore, the value of function which is based on the scores showing the level of abnormality, not the order, should be presented. The second problem is that the current detection method does not take all indices which are related to the indices of medical claim into consideration, but it puts weight only on single certain index. The single index cannot display the overall level of abnormality in each institution. In order to resolve these problems, the overall index was developed in the precedent study [6, 7] and it quantifies the level of abnormality of the medical claim pattern and unifies

individual index. It has been applied to the HIRA system since the second half of 2009. However, because the ideal scores suggested in the precedent study ranges quite widely, they have a tendency to expand the degree of abnormality. In addition, the calculation method for variance importance that was used in unifying individual index is to calculate a mean value of the weights obtained from several statistical analyses. In this study, therefore, we suggest the function made based on the precedent study but it can generate more sophisticated scores. Furthermore, in order to grant importance of variables, we suggest a methodology which uses genetic algorithm that is one of the meta-heuristic methods

II. Proposed Method

If the 'problematic institutions' is expressed in simple form in terms of medical claim, they means the institutions that show above average claim rates. Therefore, the important core concept in designing function is to focus the investigation on these institutions which show above average claims rates. In this study, we calculate the value of function in consideration of only the values which are higher than average by indices. By summing up these values according to the importance of the indexs, we divide the institutions into two groups: normal or problematic institutions.

A. Design of Scoring Function

Through the formula (1), the institutions which have above average value are given high value of function and the institutions which have below average value are given zero as the value of function.

$$P_i = \max\left(\frac{x_{ij}-\mu_j}{\sigma_j}, 0\right) \quad \textbf{(1)}$$

i means the record index and shows each institution of the data. j means the index of claim index. μ_j and σ_j means the mean and standard deviation of each claim index respectively. Therefore, zero is given as the value of function until the size of claim index reaches the mean. However, if it is larger than the mean, the higher value of function is given as the size of claim index becomes more distant from the mean, which is shown in Figure 1.

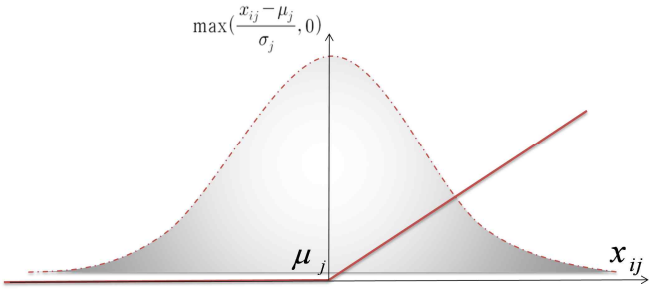


Fig.1. Function curve

Because the value of function for each claim index is obtained through the formula (1), it is necessary to sum up the values of function for all claim indices as shown in the formula (2) in order to reflect all claim indices.

$$S_i = \sum_{j=1}^{J} \alpha_j \max\left(\frac{x_{ij}-\mu_j}{\sigma_j}, 0\right) \quad \textbf{(2)}$$

The value of S_i is total value indices which presents the abnormality degree of institution and α_j is weight which presents the importance of each index (regarding the weight, we will address it in the following clause). With the value of S_i obtained through the formula (2), we design the score function as the formula (3) below in order to decide whether there is abnormality by using the critical value.

$$\hat{y}_i = \frac{2}{1+\exp\left[-(\alpha_0+\sum_{j=1}^{J}\alpha_j \cdot \max\left(\frac{x_{ij}-\mu_j}{\sigma_j}, 0\right)\right]} - 1 \quad \textbf{(3)}$$

The formula (3) is a sigmoid function and plays a role to make the value of $\hat{y}_i$ closer to binary variable by converting the value of S_i obtained through the formula (2) into the value which is located between -1 and 1 on the basis of the critical value (α_0). The obtained value of $\hat{y}_i$ is a discriminant score. If it is located close to -1, it means a normal institution. However, if it becomes closer to 1, it means an abnormal institution, a problematic institution. Figure 2 shows the division of abnormal institutions and normal institutions by the discriminant score on the basis of the critical value (α_0).

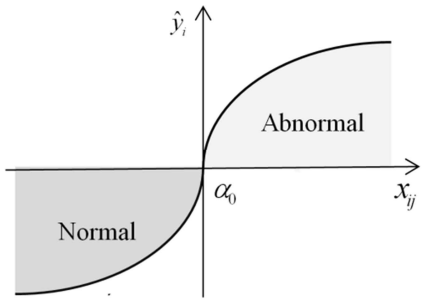


Fig. 2. Discriminant score for abnormal or normal institutions

B. Variable Weighting

In order to complete the formula (3) above, it is necessary to know the weight (α_j). At first sight, it seems that we might be able to obtain the weight by using the least-square method since the formula (3) is similar to the logistic regression analysis. However, from the formula (2), we can learn that this function cannot be differentiated. In other words, it is impossible to obtain the weight by using the least-square method. Therefore, we obtain the weight by using genetic algorithm (GA)[Davis, 1991; Holland, 1975; Goldberg, 1989]. GA performs the search process in four stages: initialization, selection, crossover, and mutation [Wong & Tan, 1994]. The initialization stage, a population of genetic structures, called chromosomes that are randomly distributed in the solution space, is selected as the starting point of the search. After the initialization stage, each chromosome is evaluated using a user-defined fitness function. The role of the fitness function is to numerically encode the performance of the chromosome. For real-world applications of optimization methods such as GAs, choosing the fitness function is the most critical step. In our study, the fitness function which is used in the space exploration process for resolution of genetic algorithm employs the Sum of Squared

Error (SSE) as shown in the formula (4). y_i is the actual value obtained from the data and $\hat{y}_i$ is the estimated value obtained from the formula (3). Under the fitness function of genetic algorithm, the chromosome which minimizes the formula (4) is chosen as the weight for variable.

$$\alpha = \arg\min \; f(\alpha) = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \qquad \textbf{(4)}$$

The overall process of genetic algorithm is shown in Figure 3 and the algorithm is presented in Table 1.

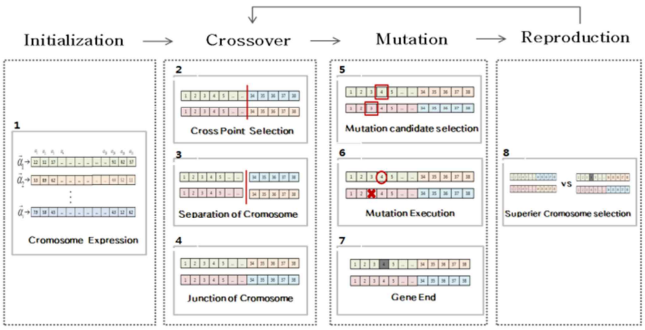


Fig.3. Genetic Algorithm application process for the exploration of variable weight

Table 1
Genetic Algorithm

Algorithm : Genetic Algorithm

begin initialize θ, P_{co}, P_{mu}, N_α, $\vec{\alpha}_i = [\alpha_1, \alpha_1, \cdots, \alpha_J]$, $\vec{\alpha}_i \in [0,10]$,

do determine fitness of each , $\vec{\alpha}_i$, f_i, $i = 1, \cdots, N_\alpha$

rank the $\vec{\alpha}_i$

do select two $\vec{\alpha}_i$ with highest score

if Rand[0,1] < P_{co} **then** crossover the pair at a randomly

else change each bit with probability P_{mu}

until N offspring have been created

until any $\vec{\alpha}_i$'s score f_i exceeds θ

return highest fitness $\vec{\alpha}_i$ (best α^*)

end

Each chromosome of genetic algorithm consists of the number of J genes which presents the importance of variable. N_α means the number of such gene. The genetic algorithm includes the process of reproduction, cross-breeding and mutation. Through these processes, the genes which have high fitness are chosen and the population is evolved. *θ, Pco*, and *Pm* means the critical acceptance value, crossover ratio, and mutation rate, respectively.

III. EXPERIMENT

The data used in the experiment was the claims data for medical expenses which was collected from the internal medicine clinics in Seoul area in the second half of 2007 and included the information of medical treatment institutions, details of claims, patients' information and information pertinent to claim settlement. The data consists of 600 which include 100 problematic institutions (y_i = 1) and 500 normal institutions (y_i =-1). The number of variable (claim index) is total 31. The value of the area under the ROC curve (AUC) was used to compare the predictive capability between methods [10]. The 5 fold Cross Validation (5-CV) was used for verification of model. With 5CV, the total data set is divided into five sets. Then four sets are used to make a model and the rest is used to verify the capability of the model. This procedure is repeated five times by applying the procedure to each set as much as possible. After repetition, the mean value is obtained as the final result value. In the following subsections, we first present the experimental results of the variable weighting with GA, and then the comparison results of the proposed method with the precedent method [6][7] and the representative data mining models.

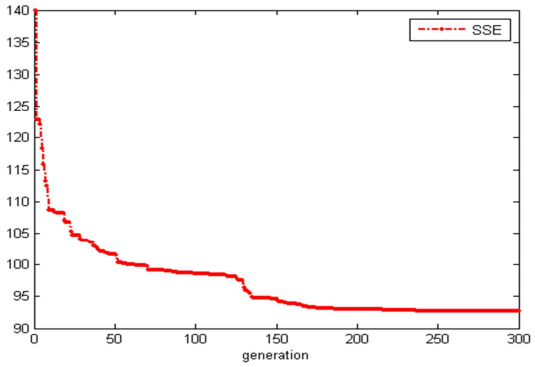
A. Variable Weighting with GA

The model parameter of genetic algorithm was set up as shown in the following Table 2.

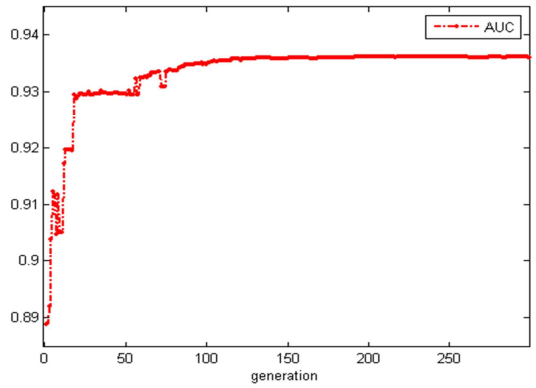
[Table 2]
Value of parameters of genetic algorithm

Parameter	
# of Pop	200
Prob. of crossover	0.80
Prob. of mutation	0.01
# of Generation	300

The genetic algorithm for established parameters explores the ideal weight by minimizing the fitness function which was designed in the previous clause. Because the weight which minimizes the fitness function maximizes the efficiency of a model, we can expect high accuracy. Figure 4 shows the convergent process of the value of fitness function (SSE) and the accuracy (AUC) as the generation proceeds. According to Figure 4(a), the SSE had continuously decreased as the generation proceeded. The SSE was above 140 at the first generation, but it decreased up to below 95 at the 300 generation. In addition, Figure 4(b) shows that the value of AUC continuously increased and that it increased from 0.89 to 0.94. This result shows that the accuracy improved as the generation proceeded.



(a) The value of fitness function (SSE) over the progress of generation

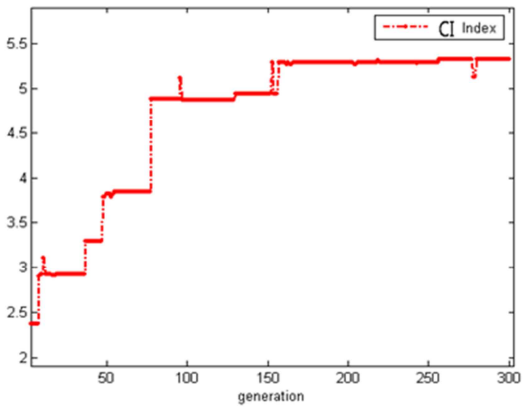


(b) AUC increase curve

Fig.4. Changes in the value of fitness function (SSE) and the accuracy (AUC) over the progress of generation

The importance of variables, in other words, the importance of assessment indices for claimed bills, is determined by the value of final chromosome gene which got through the evolution process of genetic algorithm. According to Figure 5, the weight of the index, 'CI[1)]' increased from 2.5 and then converged on near 5. The weight of the index, 'injection prescription rate,' started from 5.5. Then it continuously reduced and converged on near 2.5.

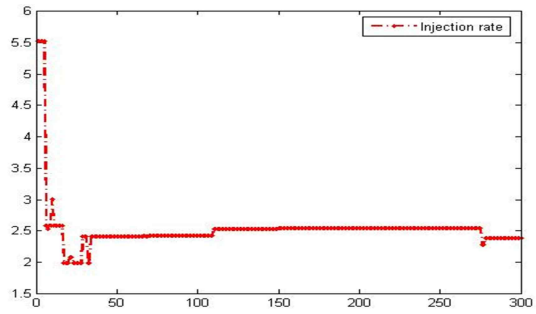
From Table 3, we can ascertain the relative value of weight of each of 31 indices which were obtained from the genetic algorithm. The size of the value means the importance of variable to explore the problematic institutions. Whereas the value of 'the visit day per case' was 1.09, it was 4.91 for 'Costliness index.' From this result, we can learn that 'Costliness index' has more important effect on exploring the problematic institutions by approximately 4~5 times than 'the visit day per case.



(a) Weight of CI

1) $$\text{Costliness Index} = \frac{\sum C_{hi} \cdot N_{hi}}{\sum C_i \cdot N_i}$$

(h: the institution, i: disease group)



(b)Weight of Injection prescription rate

Fig.5. Convergence graph of weights of CI and Injection prescription rate over the progress of generation

[Table 3]
Variable Weights
(*partial indices are not disclosed because of their confidentiality)

	Input Variables	MAD
1	Number of medicine	2.90
2	Costliness index	4.91
3	VI index	2.48
4	Medication expenses accrued outside the institution per treatment	3.04
5	Medication expenses accrued outside the institution per visit	2.29
6	CMI INDEX	1.55
7	Consultation fee CI	2.67
8	Oral administration CI	2.77
9	Psychological fees CI	2.96
10	Operation FEE CI	1.86
11	Diagnosis FEE CI	1.83
12	PET CI	1.64
13	Antibiotics prescription rate	2.01
14	Injection prescription rate	2.38
15	Medicine cost per	2.44
16	Rate of prescribed costly medicine	1.60
17	Number of medication per prescription	2.37
18	Rate of prescriptions more than 6 medicine	2.06
19	Digestives prescription ratio	1.48
20	Adrenalin Cortex-respiratory	1.30
21	Adrenalin Cortex-joint	0.76
22	Number of injury per detailed statement	2.50
23	Visit day per case	1.09
24	Administration day per case	2.55
25	Medication expenses accrued outside the institution per receiver	2.86
26	Total amount of treatment fees per receiver	2.43
27	top ranked CI	3.46
28	the 2nd ranked CI	3.10
29	the 3rd ranked CI	2.48
30	the 4th ranked CI	2.77
31	the 5th ranked CI	2.30

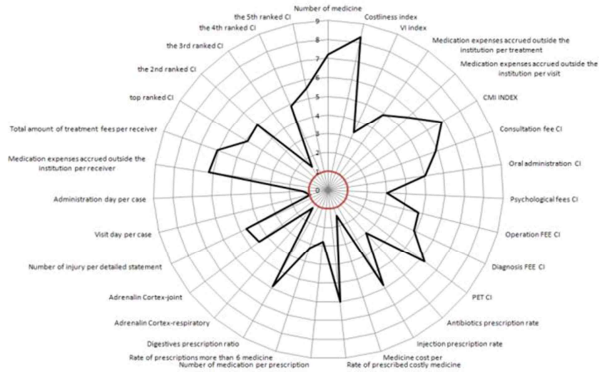


Fig.6. Diagram showing the relative importance of variable

B. *Comparison Results*

We compared the proposed method with the methods in the precedent study [6, 7] and several representative data mining algorithms. Hereafter, the method performing Medical Bill-Claim Abuser Detection is denoted as MAD for convenience. First, we compare the proposed method using the weight that was explored through the genetic algorithm (MAD_{GA}) and the method using equal weighting on the assumption that all α_j are the same (MAD_{EW}). The comparison experiment was conducted by carrying out 5-CV 15 times. The results were the mean AUC, which were shown in the second and the final rows of Table 4. From the comparison of the AUC which showed that the AUCs for MAD_{GA} and MAD_{EW} were 0.965 and 0.900, respectively, we learned that MAD_{GA} has much higher predictive accuracy. Moreover, from the comparison of the standard deviation which showed that the standard deviations for MAD_{GA} and MAD_{EW} were 0.017 and 0.008 respectively, we learned that MAD_{GA} is more stable. Since MAD_{GA} generates high accuracy and the changes in its results are stable, it is superior to MAD_{EW}.

[Table 4]
Performance Comparison with AUC Values
(MAD_{GA}, MAD_{EW}, and MAD_{PW} indicates different variable weighting methods, from GA (the proposed method in this study), from equal-weighting, and from the precedent study [6][7], respectively.)

15 5-CV	Reg	**NN**	**DT**	**MAD_{EW}**	**MAD_{PW}**	**MAD_{GA}**
1	0.923	0.907	0.669	0.912	0.933	0.968
2	0.920	0.918	0.630	0.893	0.925	0.956
3	0.885	0.746	0.759	0.904	0.922	0.968
4	0.916	0.919	0.825	0.860	0.912	0.955
5	0.888	0.906	0.646	0.924	0.903	0.982
6	0.924	0.913	0.795	0.905	0.901	0.968
7	0.900	0.904	0.749	0.896	0.936	0.970
8	0.919	0.900	0.646	0.886	0.916	0.951
9	0.862	0.908	0.669	0.918	0.922	0.968
10	0.913	0.930	0.763	0.894	0.917	0.961
11	0.882	0.901	0.693	0.899	0.880	0.972
12	0.900	0.920	0.773	0.917	0.937	0.966
13	0.871	0.881	0.749	0.898	0.922	0.975
14	0.901	0.879	0.798	0.918	0.938	0.964
15	0.886	0.897	0.749	0.879	0.929	0.958
μ	0.899	0.895	0.728	0.900	0.920	**0.965**
σ	0.020	0.044	0.063	0.017	0.016	**0.008**
p-value	**0.000** <0.05	**0.000** < 0.05	**0.000** < 0.05	**0.000** < 0.05	**0.000** < 0.05	

In the same way, we compared the weight obtained through the precedent study (MAD_{PW}) [6, 7] with the weight obtained through MAD_{GA}. The mean and standard deviation of AUC in MAD_{PW} were 0.920 and 0.016 respectively, which means that the efficiency of MAD_{PW} is short of the efficiency of MAD_{GA}. From the results comparison between 6th and 7th columns in Table 4, we can learn that the suggested genetic algorithm method is superior to the method obtained through the precedent study (MAD_{PW}).

The most prevalent methods for detection of medical fraud are Decision Tree (DT), Neural Network (NN), and regression analysis (Reg). The mean AUC for each method is shown in [Table 5]. While the means of AUC for regression analysis, neural network and decision tree were 0.899, 0.895 and 0.728 respectively, the AUC of MAD_{GA} was 0.965. Thus, the AUC of MAD_{GA} was highest among them. In addition, the standard deviation of AUC in MAD_{GA} was 0.008, which means that MAD_{GA} showed stable results compared to other algorithm. The very top of curve in Figure 7 was the ROC of MAD_{GA} and it always shows the highest value for all thresholding values. The regression analysis and neural network show the similar shape of the ROC curve. However, decision tree shows lower predictive value than other algorithms do as the thresholding value gets high while it shows high accuracy in low thresholding values.

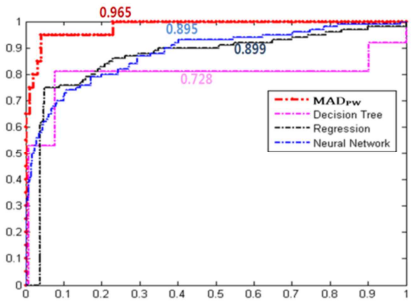
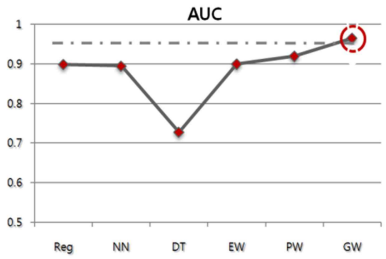
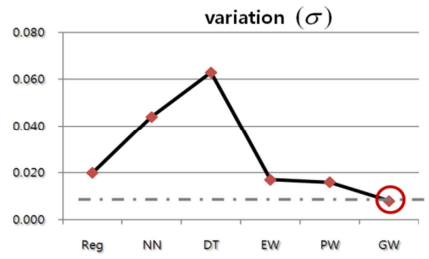


Fig.7. Comparison with ROC Curves

Figure 8 shows the mean and deviation of AUC values of different methods in Table 4. The two graphs, (a) and (b), present that the proposed method (MAD_{GA}) is the most accurate and stable than any other methods in comparison



(a) Comparison of AUC



(b) Comparison of variation

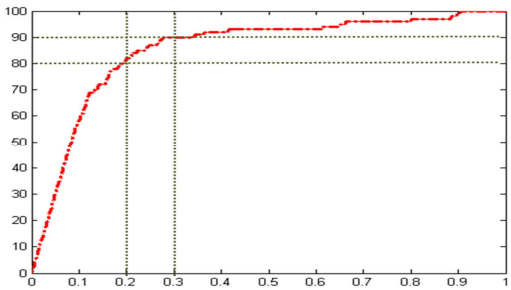
Fig.8. Comparison of accuracy and stability.

The last two rows of Table 6 present the result of t-test which tested whether there was a significant difference between MAD_{GA} and other methods. Through t-test, we can test whether the difference between AUC of other methods and that of MAD_{GA} is caused by accident or by the actual difference in efficiency. Because the p-values were near 0 which was lower than 0.05, a significance level, we can learn that the excellent efficiency of MAD_{GA} is statistically significant.

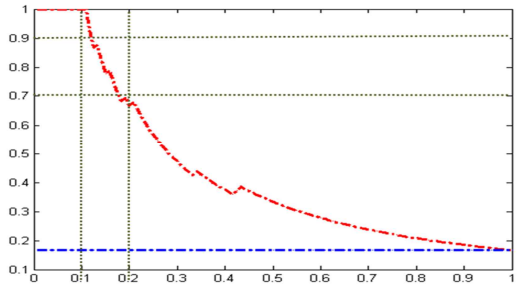
C. *Practical Implication*

Figure 9 shows Lift and Response chart which were generated through MAD_{GA}. The Y-axis in Lift chart means the number of problematic institution. In Response chart, it means the hit ratio of problematic institution. The x-axes of two charts mean the institutions which were arrayed based on the predictive values obtained from the formula. According to Lift chart in Figure 9(a), if we select the top 20% problematic institutions, we can explore 80% of the total problematic institutions. If the top 30% are selected, 90% can be explored. From Response chart in Figure 9(b), if the top 10% problematic institutions are selected, all of them are problematic institutions. If the top 20% problematic institutions are selected, 70% of the selected institutions are actual problematic institutions. After reviewing these two charts, we can interpret that if we selected 20% of institutions based on the predictive value of the suggested model, we are able to find 80% of the actual problematic institutions with 70% accuracy.

In practical setting, time and manpower generate costs. Therefore, if we use the suggested method which has high accuracy, we can expect the effect of cost reduction and improvement of efficiency because we can explore many unjust institutions with a little data.



(a) Lift chart



(b) Response Chart

Fig.9. Lift and Response charts

IV. Conclusion

In this study, we developed the medical fraud detection model by using the data in relation to medical claim for the purpose of establishing efficient national just medical expenses review system. Through the score function which is one of the suggested methods and measures the degree of abuse and unjust use of medical expenses made by health institutions focuses only on a few institutions which have a value above the mean of medical claim expenses made by all health institutions, excluding the institutions which have a value below the mean. By doing this, the score function was designed to effectively work in terms of calculation and actual applicability. Furthermore, for the determination of weight which decides the importance of the claim indices (variables) that are in relation to medical expenses overuse or abuse, we introduced the genetic algorithm. Although this method is simplified compared to the methods obtained through the precedent study [6, 7], it provide much more accurate methodology. The suggested method was compared with the precedent study, decision tree, neural network and regression analysis. The results show that the suggested model is a stable model with high predictability.

In the actual process of medical expenses review, after the problematic institutions which claims unjust or false medical expenses are selected, the affected institutions always raise issues or resist during the course of punishment procedure. Therefore, the model for selection must be accurate and the reason for selection should be clearly presented. The model suggested in this study has high predictability. Besides, the variable weight method using the genetic algorithm makes it clear which variable is important in exploring the problematic institutions and how much important the variable is. Thus, this method provides efficiency by raising the transparency in relation to the results of medical expenses review.

The establishment of efficient and effective medical fraud detection system by HIRA not only saves the medical expenses but also uses medical insurances correctly which all people have to shoulder in terms of social expenses. Therefore, we will develop much more accurate and efficient medical detection system by introducing the latest data mining method and machine learning method in the further study.

Acknowledgment

H.Shin would like to gratefully acknowledge support from Post Brain Korea 21 and the research grant from National Research

Foundation of the Korean Government (KRF-2009-0065043 /2012-0000994)

REFERENCES

[1] D. E. Goldberg, *Genetic Algorithms in Search Optimization and Machine Learning*. MA: Addison-Wesley, 1989.

[2] H. J.H., *Adaptation in Natural and Artificial System: An Introduction with Application to Biology*. Ann Arbor, MI: University of Michigan, 1975.

[3] D. L., *Handbook of Genetic Algorithms*. New York, NY: Van Nostrand Reinhold, 1991.

[4] P. L. Bartlett*, et al.*, "Learning Changing Concepts by Exploiting the Structure of Change," *Machine Learning,* vol. 41, 2000.

[5] F. Bonchi*, et al.*, "A classification-based methodology for planning audit strategies in fraud detection," in *ACM SIGKDD*, New York, NY, USA, 1999.

[6] H. He*, et al.*, "Application of genetic algorithms and k-nearst neighbour method in medical fraud detection," in *SEAL*, Springer-Verlag London, UK, 1999.

[7] H. He*, et al.*, "Application of neural networks to detection of medical fraud," *Lecture Notes in Computer Science,* vol. 1585, pp. 74-81, 1999.

[8] H. Shin and J. Lee, "How to Integrate the diverse measures for hospital fraud detection," in *INFORMS*, San diego, CA, USA, 2009.

[9] H. Shin*, et al.*, "A scoring model to detect abusive billing patterns in health insurance claims," *Expert Systems with Applications,* vol. 39, pp. 7441-7450, 2012.

[10] J. A. Major and D. R. Riedinger, "EFD: A hybrid knowledge/statistical-based system for the detection of fraud," *Risk and Insurance,* vol. 69, pp. 309-324, 2002.

[11] G. J. Williams and Z. Huang, "Mining the Knowledge Mine: The Hot Spots Methodology for Mining Large Real World Databases," *Lecture Notes in Computer Science,* 1997.

[12] V. M. and C. T., "ROC curve, lift chart and calibration plot," *AMS,* vol. 3, pp. 89-108, 2006.

[13] W.-S. Yang and S.-Y. Hwang, "A process-mining framework for the detection of healthcare fraud and abuse," *Expert Systems with Applications,* vol. 31, pp. 56-68, 2006.

How a Financial Crisis Affects Data Mining Results: A Case Study

Mary Malliaris[1] **and Anastasios G. Malliaris**[2]
[1]Information Systems & Operations Management, Loyola University Chicago, Chicago, IL, USA
[2]Economics Department, Loyola University Chicago, Chicago, IL, USA

Abstract *- Successful trading of currencies over time often depends on a strategy that remains consistent. Identifying patterns that occur among such currencies is a basis for the strategy. However, major financial crises can cause shifts in trading patterns that interfere with even the best approaches. This paper uses the association analysis data mining technique to compare rules related to eight major currencies and their co-movements before and after the financial crisis of October 2008. The currencies included in the search for rules are the Australian dollar, the Japanese yen, the euro, the Swiss franc, the British pound, the Canadian dollar, the Mexican peso, and the Brazilian real. Some of the rules that remained stable, during a seven-year period, on both sides of the 2008 financial crisis are examined and compared.*

Keywords: currency patterns, association analysis, financial risk, trading rules

1 Introduction

As globalization in investments has increased, the volume of currency transactions has also increased. Because information can be accessed instantly from anywhere in the world, it is possible to follow movement of any currency from wherever you are. Funds can easily be transferred from one market to another in order to follow attractive opportunities for investment.

Globalization in the financial sector also allows for the opportunity to diversify a portfolio and thus reduce risk. The process of financial globalization, and it effect on investors, has been discussed in Lane and Milesi-Ferretti [6, 7] and Campbell et al. [3]. The growth of international financial markets has been a dominant feature of the last decade and has affected monetary policy in many countries.

Price stability has become a featured goal of monetary policy as many central banks, either explicitly or implicitly, try to form policies that target inflation. Individual investors desire to hold currencies that are negatively correlated with equities so as to minimize portfolio variance. Campbell et al. discuss strategies for combinations of currencies and equities that minimize risk.

The markets that dominate all others in terms of trading volume are those dealing with currency transactions. Certain global currencies have been the subject of much study during the last decade. As the euro replaced the German mark, the French franc, the Italian lire and others, there began a search for pricing the euro against other currencies and for understanding how its movement correlated with remaining currencies. Since currencies are priced in terms of one another, this becomes an even greater challenge to understand the inter-relationships. Pricing currencies can be viewed as a comparison of economic fundamentals between multiple nations.

Rather than focusing on a model that prices some single currency, the focus of this paper is to inspect the way that eight global currencies move together. In order to build a portfolio that is diversified, it is helpful to isolate currencies that move often in the same direction, in opposite directions, or that have no relationship to each other. Rather than prices, this paper looks at direction of movement only. The global financial crisis that involved a crash in October 2008 provides a separation point around which the data set is divided. Directional currency movement before this time and after this time is analyzed and compared. Though the crash initially affected credit markets, it quickly expanded to include equity, bond and currency markets. After Lehman Brothers filed for bankruptcy protection, the crisis amplified, and currency markets were subjected to significant volatilities. Our main goal is to see whether or not we can find relationships among currencies that remained in place both before and after this major shock to the markets. If price movements are not random, but have stable relationship patterns even through shocks, then this information can help investors to form long-term trading strategies.

Empirical evidence of chaotic dynamics in financial data such as stock market indexes, foreign currencies, and macroeconomic time series has been found by various researchers such as Kyrtsou and Vorlow [5] recently, and in much more detail earlier by Brock, Scheinkman and LeBaron [9] and Brock and Malliaris [2]. A detailed portrayal of the presence of non-linear determinism in financial markets can be located in Mandelbrot and Hudson [8].

However, this paper will focus on co-movements in eight specific currency markets before and after October 2008, and

investigate whether we can find any stable pattern on each side of a period of instability. Or, phrasing this problem in terms of the methodology, are there any Apriori rules for directional movement of the eight major currencies that are common to the years prior to and following October 2008?

2 Data and Methodology

Each business day at 12:30 ET, the Bank of Canada published the nominal noon exchange rates for each of the eight currencies used in this study. These currencies are: the Australian dollar, the Japanese yen, the Euro, the Swiss Franc, the British Pound, the Canadian Dollar, the Mexican Peso, and the Brazilian Real. The source site for the data is http://www.bankofcanada.ca/rates/exchange/10-year-converter/. Each original number is the amount of the currency equal to one US dollar on that day at that time. The data covered a time period beginning in November 2005, and ending in September 2011.

In early October of 2008, the Dow Jones fell over 18% and the S&P 500 fell more than 20%. Other world markets followed this crash by also registering declines. The month of October 2008 was a period of major instability and is thus a dividing time in our data set. Removing this month divided the data set into two distinct pieces which we label Before (October 2008) and After. Each of these two data sets contains over 700 days of data. The Before and After sets were further subdivided into training and validation sets with the validation set being the last 252 days of each set. The validation sets thus occur entirely after the training sets and are completely disjoint. This type of disjoint, temporally following, and lengthy validation set is the most difficult for a model to perform well on and will thus be a very good judge of the rules stability. The training sets were used to generate rules of directional movement for the currencies. The validation sets were used to judge the robustness of these rules on entirely new data. The four sets are named Before Training [Bef Tr], Before Validation [Bef Val], After Training [Aft Tr], and After Validation [Aft Val]. The beginning and ending dates, along with the number of rows, for each set are shown in Table 1 along with a visual timeline.

Table 1. Data Sets

Set	Begin	End	Size
Before Tr	11/1/2005	9/28/2007	480
Before Val	10/1/2007	9/30/2008	252
After Tr	11/3/2008	9/17/2010	471
After Val	9/20/2010	9/19/2011	252

Before Tr	Bef Val		After Tr	Aft Val

The data was originally downloaded as numeric values and converted into category-type data that represented the direction of movement from the previous day of the respective currency relative to the US dollar. The converted values thus were Up and Down. Since all the data points are generated by same-time prices, it is useful to look at the number of Up movements common to the eight currencies over the four data sets. These are shown numerically in Table 2, and graphically in Figure 1.

Table 2. Values Shown as Percent of Total Number of Days

Numb. Ups	Bef Tr	Bef Val	Aft Tr	Aft Val
0	7.50	6.35	8.92	11.95
1	11.88	13.10	15.71	11.95
2	14.38	16.67	7.64	13.94
3	13.96	12.30	12.53	12.35
4	12.71	12.30	11.25	9.96
5	11.88	12.70	10.19	13.94
6	12.92	9.92	14.23	10.36
7	10.83	11.11	14.44	7.57
8	3.96	5.56	5.10	7.97

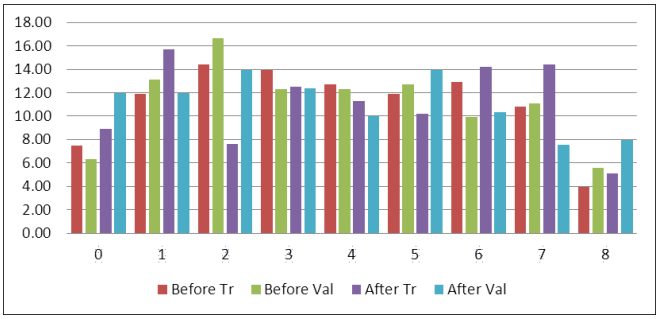


Figure 1. Comparison of Percent Ups

In particular, notice the numbers for 0, 4, and 8. In 0 and 8, all the markets move together, and this has increased since the crash. In 4, half the markets move one way, and half the other. This percent has decreased over this time period. With the increasing flow of information made possible in today's world, it seems that markets are more likely to agree than not. However, when they do not, then data mining can help us to understand the markets that do and do not move together.

Though association analysis originated with the study of market baskets to see which items people purchased at the same time, it has been generalized to look at questions of what occurs together. It is often used in an exploratory way to discover interesting relationships in the data that may be

analyzed further. For an in-depth discussion of association analysis techniques, see Hand et al [4], or Berry and Linoff [1].

The methodology used to generate rules on these currencies was Apriori Association Analysis, run in IBM's SPSS Modeler data mining package. Association rule algorithms automatically find data patterns that you could find by manually counting, but with much greater speed. Apriori extracts a set of rules from the data set, pulling out the rules with the information content meeting requirements specified by the user. The rules generated by this methodology associate each particular outcome with a specific set of conditions. The technique does not limit us to one outcome only, but can consider each of the variables in turn as either possible antecedents or consequents. Thus, the methodology used seven of the currencies as possible inputs and one currency as output. It then repeated the process, varying the set of input and output variables so that each currency and direction had the chance to be considered as a consequent. Association rules are not used directly for predicting, but are useful for understanding various patterns that have occurred in the data set. The Apriori technique is fast and efficient. It has no arbitrary limit on the number of rules that can be retained and it can handle rules with up to 32 antecedents.

Association analysis generates a set of rules of the form IF A THEN B where the variables used in the modeling process may occur either after the IF or after the THEN. In this study, there were eight currency variables, and each could have one of two possible values, Up or Down. The IF part of the rule could use any combination of markets and directions. The THEN part contains only one market with one direction and does not appear as part of the IF clause.

The set of rules that is generated also depend on the user-supplied minimum values of support and confidence. Support refers to the percent of times that some combination of inputs (also called antecedents) occurs in the data set. That is, it tells us the percent of rows for which the antecedents are true, based on the training data. This forms the IF part of the statement. When the antecedent combination does occur, confidence reflects the percent of time that the output, or consequent, is also true. This is the confidence of the THEN part of the rule. In this problem, support and confidence were set to minimal values of 7% and 65%, respectively. Thus for any rule to be generated, the IF part must occur in the training data set at least 7% of the time, and when the IF part is there, the THEN part must be true at least 65% of the time. Setting these values lower generates more rules, and setting them higher causes fewer rules to be located. There is no fixed value commonly used, and these will depend on how important the frequency and accuracy of occurrence is to the end-user. The IF part of the rule can use anywhere from one to seven markets. The conbinations with more antecedents typically have lower support and higher confidence than the same rule would with one less antecedent.

IBM's SPSS Modeler is a data mining package with numerous models and drag-and-drop construction. Each part of the data mining process is represented by a node that appears in the data stream on the model pallette. A picture of this process is shown in Figure 2.

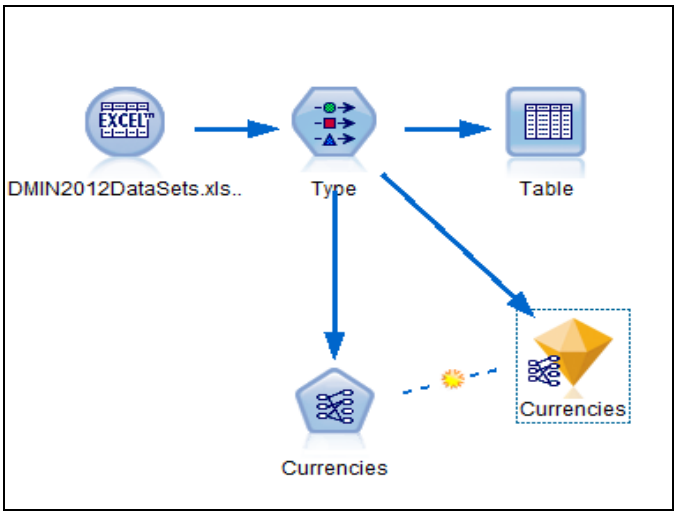
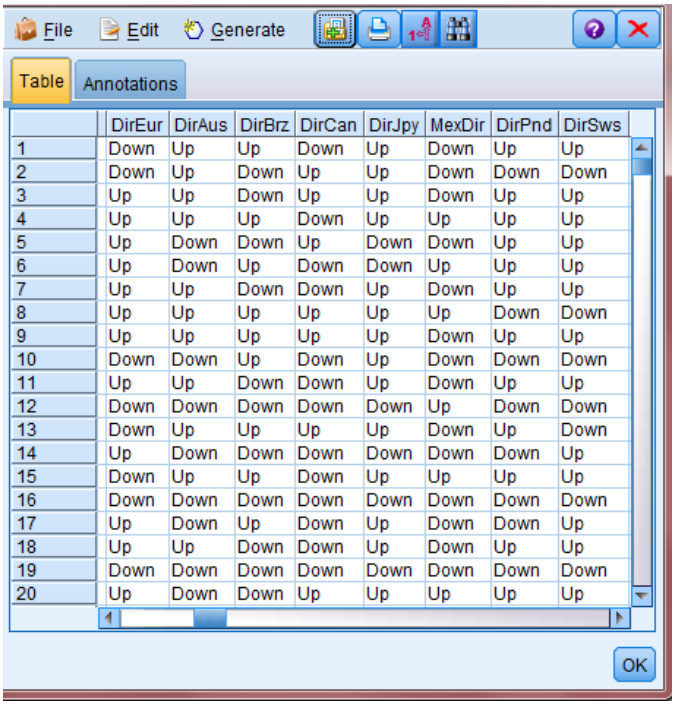


Figure 2. Data mining stream in Modeler.

In this figure, the data from an Excel spreadsheet is fed into the model. The data then flows into a Type node where the data type and use for each field is identified. Data can be viewed in the Table node to ensure that it is being viewed correctly by Modeler [see Figure 3].



	DirEur	DirAus	DirBrz	DirCan	DirJpy	MexDir	DirPnd	DirSws
1	Down	Up	Up	Down	Up	Down	Up	Up
2	Down	Up	Down	Up	Up	Down	Down	Down
3	Up	Up	Down	Up	Up	Down	Up	Up
4	Up	Up	Up	Down	Up	Up	Up	Up
5	Up	Down	Down	Up	Down	Down	Up	Up
6	Up	Down	Up	Down	Down	Up	Up	Up
7	Up	Up	Down	Down	Up	Down	Up	Up
8	Up	Up	Up	Up	Up	Up	Down	Down
9	Up	Up	Up	Up	Up	Down	Up	Up
10	Down	Down	Up	Down	Up	Down	Down	Down
11	Up	Up	Down	Down	Up	Down	Up	Up
12	Down	Down	Down	Down	Down	Up	Down	Down
13	Down	Up	Up	Up	Up	Down	Up	Down
14	Up	Down	Down	Down	Down	Down	Down	Up
15	Down	Up	Up	Down	Up	Up	Up	Up
16	Down	Down	Down	Down	Down	Down	Down	Down
17	Up	Down	Up	Down	Up	Down	Down	Up
18	Up	Up	Down	Down	Up	Down	Up	Up
19	Down	Down	Down	Down	Down	Down	Down	Down
20	Up	Down	Down	Up	Up	Up	Up	Up

Figure 3. Data viewed in the Type node.

The data from the Type node then flows to the Apriori node [labeled Currencies at the bottom]. Within the Apriori node, the user can set the limits for support and confidence for the specific problem as shown in Figure 4.

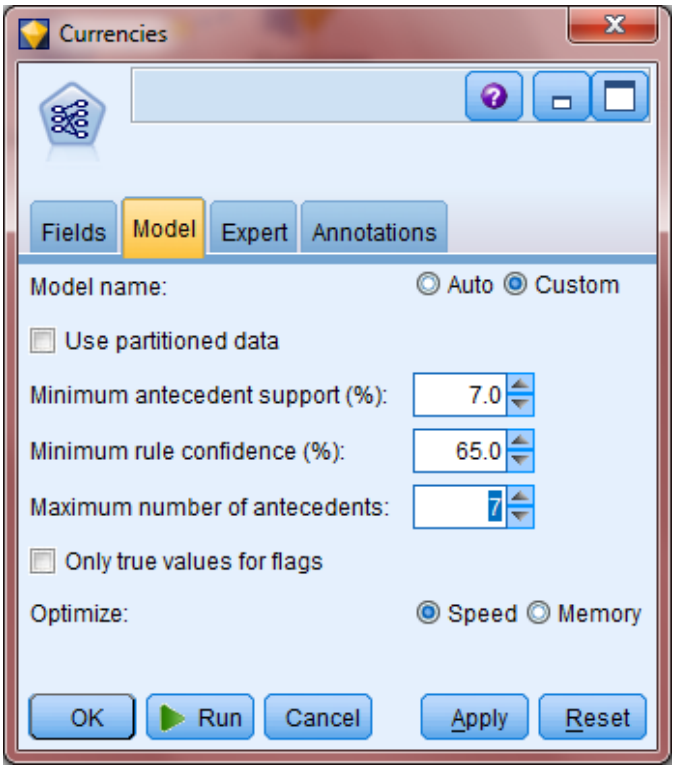


Figure 4. Apriori Model settings

After running the Apriori model, a generated nugget containing the results is attached to the right side of the model. Right-clicking the generated model allows the user to browse the set of rules that has been created. These rules can be filtered to search for results relating to a specific input or target. They can also be sorted by support or confidence if the user is looking for the most reliable or most frequently occurring rules. Once a trained model is created, other data sets can be attached to the Type node and few through the generated model to see what rules are used by the new data.

3 Apriori Results

Using the settings for support and confidence detailed above, the Before training set generated 2635 rules. The After training set generated 2643 rules. While there are many ways to look at the results from these training runs, we will focus on those rules that occurred in both of the training sets. These are the rules that have remained stable on both sides of the October 2008 crash.

Of these two large rule sets built on the training data, 79 rules had identical antecedents and consequents. That is, 79 common rules of the form IF [antecedent] THEN [consequent] occured in both the Before and After training sets. From these 79, eleven rules with various markets and directions were selected for further analysis in this paper. [Not every market and direction combination generated a rule common to both sets.] These eleven common rules are shown in Table 3. The antecedents contain multiple conditions, all of which must be true for the rule to be applicable. For example, Rule 1 states that if the Japanese yen was Down (at 12 noon ET, relative to the day before and relative to the US dollar) and the Mexican peso was Down and the Brazilian real was Down then the Australian dollar was also Down. Of these rules, only rule 7 used four antecedents. Rules 4 and 9 have two antecedants. All the remaining rules have three antecedents each.

Table 3. Eleven rules common to Before and After

Rule	Antecedent	Consequent
1	DirJpy = Down and DirMex = Down and DirBrz = Down	DirAus = Down
2	DirMex = Up and DirSws = Down and DirAus = Down	DirEur = Down
3	DirBrz = Up and DirSws = Up and DirMex = Down	DirEur = Up
4	DirMex = Up and DirSws = Down	DirJpy = Down
5	DirAus = Up and DirSws = Up and DirMex = Down	DirJpy = Up
6	DirSws = Down and DirCan = Down and DirAus = Down	DirPnd = Down
7	DirAus = Up and DirCan = Up and DirSws = Up and DirMex = Down	DirPnd = Up
8	DirBrz = Up and DirMex = Up and DirEur = Down	DirSws = Down
9	DirEur = Up and DirMex = Down	DirSws = Up
10	DirSws = Up and DirCan = Down and DirAus = Down	DirMex = Down
11	DirBrz = Up and DirCan = Up and DirSws = Up	DirMex = Up

Each of these eleven rules occurred in both the Before and After training sets, but with different values for support and confidence.

Though the minimum values for these are set at run-time, the Modeler package calculates the actual values of support and confidence for each rule that occurred in the data set used for training.

These actual values for support and confidence are shown in Table 4. Notice that all confidence values are above 70%. This means that, when the If part of the rule was satisfied, the THEN part was true at least 70% of the time. A change in support before and after October 2008 is an indication that the combination used in the rule is occurring more (or less) often. But as long as the confidence remains high, the rule will be accurate when it is triggered.

Table 4. Values of support and confidence for Training Sets

Rule	Bef Sup	Bef Conf	Aft Sup	Aft Conf
1	15.63	84.00	17.20	79.01
2	15.83	92.11	7.86	81.08
3	7.29	85.71	9.13	90.70
4	23.75	74.56	17.41	70.73
5	14.17	88.24	9.98	70.21
6	26.25	84.13	28.03	82.58
7	7.50	94.44	7.01	78.79
8	15.21	89.04	8.07	84.21
9	23.13	91.89	19.32	82.42
10	9.58	76.09	11.04	75.00
11	14.17	75.00	23.57	75.68

Table 5. Values of support and confidence for each validation set.

Rule	Bef Sup	Bef Conf	Aft Sup	Aft Conf
1	16.67	80.95	25.50	93.75
2	14.29	86.11	9.16	56.52
3	9.52	87.50	6.37	87.50
4	25.40	93.75	24.30	85.25
5	13.49	67.65	7.57	78.95
6	23.81	80.00	24.70	70.97
7	8.73	95.45	2.39	66.67
8	12.30	100.00	7.17	77.78
9	24.21	88.52	15.94	60.00
10	11.51	89.66	14.34	86.11
11	19.05	68.75	15.54	89.74

However, a much harder test is the comparison on each validation set. The validation set data is from the year following the training set in each of the two cases. These results on the respective Validation sets are shown in Table 5. In looking at the contrast between the results from the training and validation sets, the two greatest differences in confidence from the Before sets come from rules 4 and 5. These rules both involve the Swiss franc and the Mexican peso and have and antecedent of the Japanese Yen. Thus, there was a change in the year before the crash that affected this combination of markets.

In comparing the training and validation sets for the two After crash data sets, we see that the greatest change in confidence occurred in rules 2 and 9. These rules all involved some combination of the Mexican peso, the Swiss franc and the Euro. Thus, as we move further away from the crash, we see this relationship combination changing.

Combining the results from all four data sets, we see that the greatest changes in confidence of the rules involved the Swiss franc and the Mexican peso and their relationships to other currencies. Many of the other rules, however remained robust.

4 Conclusions

This paper considers eight major foreign currency markets and uses association analysis to look for currency rules that occur with respect to directional movements of these markets when they are all priced relative to the U.S. dollar. These directional movements all occur at the same point in time and give us a good picture of which markets tend to move Up together, Down together, or consistently in opposite directions. In addition, the lack of a rule relating any two markets is an indication that there has not been simultaneous movement in the past. Thus they may move independently in the future.

Identified patterns in these currency markets can be used as investment, hedging and speculative aids to reduce risk. Risk can be reduced when we can identify certain fundamental relationships that exist between and among currencies and that persist despite the surrounding uncertainty caused by a financial crisis. If the association analysis rules indicate that certain markest typically move together, and we see that is not happening today, then this may be an indication that a correction is likely to occur in one of those markets soon. Also, if a rule shows us that some currencies move in identifiable related patterns, then we would not look for diversity of movement when owning these currencies. A diversified portfolio would lead us to buying currencies for which no rule can be identified that relates their movements, whether Up or Down.

To uncover currency relationships in these time periods around the crash of 2008, and to examine their robustness over time and across various currencies, an Apriori association

analysis was performed on two sets of data, prior to the October 2008 financial crisis, and after. As mentioned in the results, of the initial 2635 rules, only 79 were repeated after the crash of October 2008. This is an indication that a financial crash can disrupt old patterns and cause new ones to form. Of the 2643 rules generated after the crash, given that 79 were repeated rules, this means that 2564 rules had not occured prior to October 2008.

There are eleven rules based on the eight currencies studied that this paper identifies as most appropriate for further analysis. All eleven rules, if analyzed one at a time, confirm stable relationships among currencies that would allow hedging and speculative activities. Looking, in particular, at rule 3, we see the least amount of change in confidence among the four sets of data. Though the rule occurs in amounts varying from 6.37% to 9.52%, all confidence levels are between 85 and 90%. This rule state that If the Swiss franc and Brazilian real are Up and the Mexican peso is Down, the the Euro will be Up at the same point in time.

Rather than having all currencies rising or all declining, we have eleven mixed case rules that are identified here. For example if the Swiss franc and Australian dollar decrease while the Mexican Peso increases the rule suggests that the euro will also decline. Overall, it is confirmed in these rules that the Swiss Franc and the Mexican Peso frequently move in opposite directions; also it is verified that the Mexican peso and Brazilian real often move together. The British pound is influenced by, and moves with, the Australian dollar.

In conclusion, these rules demonstrate that stable and robust relationships exist among groups of currencies that in turn form the fundamentals for global banking and investment, hedging and speculative activities. Association analysis has found patterns that have occurred in data sets both before and after the financial crash of October 2008. Of the seventy-nine configurations found in all data sets, eleven example patterns were shown here and demonstrate that data mining can be used to identify currencies whose movement, whether in the same or opposite directions, follows a consistent blueprint over time.

5 References

[1] Berry, M. and Linoff, G. Data Mining Techniques, Second Edition, Indianapolis, IN: Wiley Publishing Inc., 2004.

[2] Brock, W and Malliaris, A. Differential Equations, Stability, and Chaos In Dynamic Economics, Amsterdam, Netherlands: Elsevier Science, 1989.

[3] Campbell, JY, Medeiros KS-de, Viceira LM. *Global Currency Hedging*. Journal of Finance, 2010, 65(1):87-122.

[4] Hand, D., Mannila, H., and Smyth, P. Principles of Data Mining, Cambridge, MA: The MIT Press, 2001.

[5] Kyrtsou C and Vorlow C. *Modelling Nonlinear Comovements Between Time Series*. Journal of Macroeconomics, 2009, 31(1): 200–211.

[6] Lane, P., and Milesi-Ferretti, G.M. *The External Wealth of Nations: Measures of Foreign Assets and Liabilities for Industrial and Developing Countries.* Journal of International Economics, 2001, 55, pp. 263-94.

[7] Lane, P., and Milesi-Ferretti, G.M. *The External Wealth of Nations Mark II*. IMF Working Paper, No 06-69, 2006.

[8] Mandelbrot, B. and Hudson, R. The (Mis)Behavior of Markets. New York, NY: Basic Books, 2004.

[9] Scheinkman, Jose A and LeBaron, B. *Nonlinear Dynamics and Stock Returns*. Journal of Business, University of Chicago Press, 1989, 62(3), pages 311-37.

Constrained Nonnegative Matrix Factorization based Feature Selection

Nirmal Thapa and Jun Zhang

***Abstract*—Feature selection has been a key area of research in the data mining community for quite some time and many interesting algorithms have been developed. In this paper, we propose a novel approach to feature selection using Nonnegative Matrix Factorization (NMF) combined with traditional feature selection procedures. We put forward our motivation for using Constrained Nonnegative Matrix Factorization (CNMF) for feature selection. We propose a two stage procedure in which the first stage is to perform matrix factorization and the second phase is the traditional feature selection procedure. Our method supplements traditional methods that result in efficient technique for feature selection still maintaining accurate results in terms of clustering and classification. We test the accuracy of our method with that of traditional methods. We experiment with K-means for clustering results. Efficiency of our methods can be realized if we need to do the discretization before the feature selection. We present extensive experimental results to show the efficiency of our method.**

Keywords: Constrained Nonnegative Matrix Factorization, Feature Selection.

I. Introduction

Data with high dimension have always been a problem as far as data mining is concerned. They are hard to work on, because more features increase noise, resulting in insufficient observations. Irrelevant and redundant features adversely affect the machine learning algorithms. Algorithms like Nearest Neighbor, Decision Tree, Naive Bayes are sensitive to irrelevant attributes [6], [7], [9], [8]. One solution to eliminating irrelevant features is to perform Data Preprocessing [14], which is an essential data mining procedure.

Among others techniques to preprocess data, Feature Selection (FS) addresses the issue of dimension reduction[15], [13]. In [2] Peng et al., defined feature selection as: Given the input data D tabled as N samples and M features X=$\{x_{i:i=1,...,M}\}$, and the target classification variable c, the feature selection problem is to find from the M-dimensional observation space, R^M, a subspace of m features, R_m, that "optimally" characterizes c.

Reduction in dimension brings efficiency in terms of storage costs, computation costs, classification performance, and ease of interpretation. Feature selection methods search through the subsets of features and find the best one among the competing 2^N candidate subsets according to some evaluation function. Other methods based on heuristic or random search methods attempt to reduce computational complexity by compromising performance.

Within the data mining field, matrix decomposition has been used in obtaining some form of simplified low-rank approximation to original data. Decomposition reveals the structure of the data and makes it easier to observe relationships with the subjects and within attributes [4]. We propose novel feature selection technique that combines Nonnegative Matrix Factorization with the feature selection methods. The main part of our research is the evaluation and comparison of our method with traditional feature selection methods. We measure accuracy as well as the execution time to compare methods. We seek answers to the following questions;

- How does accuracy of the proposed method compare with the accuracy of the traditional method?
- How much computational cost does our method require, when compared with the traditional method?

The remaining part of the paper is organized as follows. Section II offers background on K-means, Nonnegative Matrix Factorization, and a couple of feature selection algorithms. Section III offers the contribution of the paper. The idea of using the CNMF is proposed and the time complexity of the methods proposed are discussed. Section IV discusses the experimental setup i.e., the datasets used for our experiments, the results observed. Finally, we conclude our work with a summary of results and present probable direction for future work.

II. Background and related work

A. K-means Clustering

Clustering is an essential part of data mining and often used as an evaluation criterion for experiments using feature selection with the machine learning algorithms. Feature selection is considered successful if the dimensionality of the data is reduced and the accuracy of a learning algorithm improves or remains the same. K-means which is the most popular clustering algorithm is chosen for our study.

The basic objective of K-means is to cluster the n data items that can be given by $(x_1, x_2, ..., x_n)$, into k sets $(k \leq n), S = (S_1, S_2, ..., S_k)$ so as to minimize the within-cluster sum of squares. Euclidean distance is a common metric and variance is used as a measure of cluster scatter.

B. Data Discretization

Discretization is the process of converting real number data into a typically small number of finite values. Many feature selection algorithms [16], [18], [19], [20] are shown

Nirmal Thapa is with Department of Computer Science, University of Kentucky, Lexington, KY, 40506, Tel: (859) 227-6786, Fax: (859) 323-1971, Email: nirmalthapa@uky.edu

Jun Zhang is Professor in Department of Computer Science, University of Kentucky, Lexington, KY, 40506, Tel: (859) 257-3892, Fax: (859) 323-1971, Email: jzhang@cs.uky.edu

to work effectively on discrete data or even more strictly, on binary data. We show that our method is very effective when it comes to feature selection with discretization. For our experimental purpose we test our experiments mainly on three different discretization algorithms: CAIM [21], Ameva [23], and CACC [22].

C. Nonnegative Matrix Factorization

Recently NMF has been shown to be very useful in approximating high dimensional data with nonnegative components [11], [5]. NMF imposes additional constraint that none of the elements of both the factor matrix H and basis matrix W can be negative. In addition, another notable property of NMF is that factorized matrix are non-unique.

Nonnegative Matrix factorization is a way in linear algebra where a matrix A is decomposed into the product of two matrices.

$$NMF(A) \Rightarrow H \times W$$
$$A \approx H \times W = A^*$$

Formally it can be defined as *Given a nonnegative data model* $A(n \times m)$*, find two nonnegative matrices* $H^{n \times k}$ *and* $W^{k \times m}$ *with* k *being the number of clusters in* A*, that minimize* Q*, where* Q *is an objective function defining the nearness between the matrices* A *and* HW*. The modified version of* A *is denoted as* A^**=* $H \times W$ *.*

There are two main aspects of NMF, one is the $update\ rule$ and the second one is the $cost\ function$. The Euclidean distance or the Frobenius norm is the commonly considered cost function:

$$\min_{H \geq 0, W \geq 0} f(A, H, W) = \|A - HW\|_F^2$$

D. Constrained Nonnegative Matrix Factorization

Many kinds of constraints can be imposed on NMF. NMF with additional constraints like orthogonality constraint [11], sparseness constraint [12] have been applied in various fields. In our previous work [10], we tried to improve pattern hiding by imposing additional constraint called $Clustering\ Constraint$. We use matrix C referred as $Clustering\ Matrix$ to define the desired clustering of the resultant matrix, it will be based on the results from k-means. The process will result in the factorization of matrix A into H and W where, W will very closely represent the clusters and H will represent the clustering result.
The objective function can be modified to accommodate penalty terms as;

$$f(A, H, W) = \alpha \|A - HW\|_F^2 + \beta \|H - C\|_F^2 \quad (1)$$

Here, C is a matrix of size $n \times k$, and the elements of C are such that

- element=1, if index of element represents the cluster to which the item belong.
- element=0, if index of element does not represent the cluster to which the item belong.

Typical example of it is

C1 C2 C3

$$\begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{matrix} \text{Item in Cluster 3} \\ \text{Item in Cluster 1} \\ \text{Item in Cluster 2} \end{matrix}$$

E. Update Formula

Mathematical derivation for update formula

Let,

$$\begin{aligned} Q &= \|A - HW\|_F^2 \\ &= tr((A - HW)^T (A - HW)) \\ &= tr(A^T A - A^T HW - W^T H^T A + W^T H^T HW) \\ &= tr(A^T A) - 2tr(A^T HW) + tr(W^T H^T HW) \quad (2) \end{aligned}$$

also let,

$$\begin{aligned} L &= \|H - C\|_F^2 \\ &= tr((H - C)^T (H - C)) \\ &= tr(H^T H - H^T C - C^T H + C^T C) \\ &= tr(H^T H - 2H^T C + C^T C) \quad (3) \end{aligned}$$

- H fixed and W changing,

$$\begin{aligned} &\frac{\delta f(A, H, W)}{\delta W} \\ &= \frac{\delta(\alpha \|A - HW\|_F^2 - \beta \|H - C\|_F^2)}{\delta W} \\ &= \alpha \frac{\delta(Q)}{\delta W} - \beta \frac{\delta(\|H - C\|_F^2)}{\delta W} \\ &= -2\alpha \frac{\delta(tr((A^T HW)))}{\delta W} + \alpha \frac{\delta(tr((W^T H^T HW)))}{\delta W} \\ &= -2\alpha H^T A + 2\alpha H^T HW \quad (4) \end{aligned}$$

- W fixed and H changing

$$\begin{aligned} &\frac{\delta f(A, H, W)}{\delta H} \\ &= \frac{\delta(\alpha \|A - HW\|_F^2 - \beta \|H - C\|_F^2)}{\delta H} \\ &= \alpha \frac{\delta(Q)}{\delta H} - \beta \frac{\delta(\|H - C\|_F^2)}{\delta H} \quad (5) \end{aligned}$$

We know, first term gives,

$$\alpha \frac{\delta Q}{\delta H} = -2\alpha AW^T + 2\alpha HWW^T \quad (6)$$

Second term gives,

$$\begin{aligned} \beta \frac{\delta \|H - C\|_F^2}{\delta H} &= \beta 2H - 2C + 0 \\ &= 2\beta H - 2\beta C \quad (7) \end{aligned}$$

Combining (6) and (7) in (5),

$$\begin{aligned} &= -2\alpha AW^T + 2\alpha HWW^T + 2\beta H - 2\beta C \\ &= 2\alpha HWW^T + 2\beta H - 2\alpha AW^T - 2\beta C \quad (8) \end{aligned}$$

For optimal solution $\frac{\delta q}{\delta W}$=0 and $\frac{\delta q}{\delta H}$=0. Hence,

$$H^T A \oslash H^T HW = I$$

$$H(\alpha WW^T + \beta) \oslash (\alpha AW^T + \beta C) = I$$

where, $\oslash$ represents element-wise division, I denotes identity matrix. This gives rise to the update formula for W and H as,

$$W_{i,j} = W_{i,j} \frac{[H^T A]_{i,j}}{[H^T HW]_{i,j}} \tag{9}$$

$$H_{i,j} = H_{i,j} \frac{[\alpha AW^T + \beta C]_{ij}}{[H(\alpha WW^T + \beta)]_{ij}} \tag{10}$$

This is the derivation, which we utilized in our previous work for data privacy [10].

F. Feature Selection Algorithms

In this paper, we mainly deal with two traditional feature selection algorithms, they are;

1) Feature Selection Based on Pearson's Correlation: From works of [25], [26], [27], correlation relation between a composite test consisting of the summed components and the outside variable can be predicted from

$$r_{zc} = \frac{k\bar{r_{zi}}}{\sqrt{k + k(k-1)\bar{r_{ii}}}} \tag{11}$$

where r_{zc} is the correlation between the summed components and the outside variable, k is the number of components, $\bar{r_{zi}}$ is the average of the correlations between the components and the outside variable, and $\bar{r_{ii}}$ is the average inter-correlation between components. Equation 11, in fact is Pearson's correlation, where all variables have been standardized.

Alternative formula for the sample Pearson correlation coefficient is also available:

$$R_{xy} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\sqrt{n\sum y_i^2 - (\sum y_i)^2}} \tag{12}$$

where x_i and y_i are the values of i^{th} subject for attributes x and y respectively.

For our research purpose we use Greedy Forward Selection (GFS) for the Correlation based Feature Selection (CFS). It starts with the empty set and greedily adds attributes one at a time until all attributes have been added. At each step feature selection procedure adds the attribute that, when added to the current set, yields the learned structure that generalizes best. Once an attribute is added FS cannot later remove it [28].

2) Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy: The other feature selection method that we use in our paper is the minimal-redundancy-maximal-relevance criterion (mRMR) as proposed by Peng et al. in [2]. The main idea is to select the feature based on maximal statistical dependency criterion, but due to difficulty in calculating the dependency the equivalent form is used

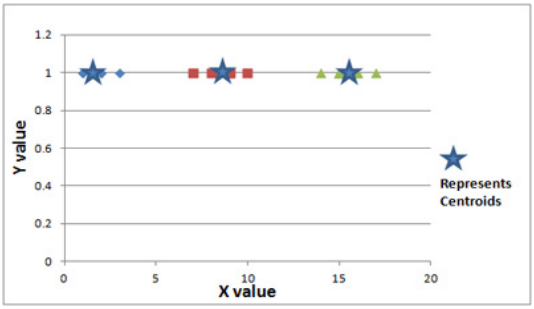


Fig. 1. Clusters of 2D data

III. Contribution of the Paper

Nonnegative matrix factorization results in dimension reduction. Matrix H and W with smaller dimension are easier to store and manipulate than matrix A. According to [17], each element H_{ij} of matrix H represents the degree to which subject i belongs to the cluster j, while each element W_{ij} of matrix W indicates to which degree attribute j is associated with cluster i. It points out that Matrix W is the compact representation of data as far as the relations between attributes and clusters are concerned. This paper aims to exploit this basic interpretation of matrix W to perform feature selection.

Further, Wang et al. [3] tried to apply this idea for data pattern hiding, they find out cluster membership of data by finding the largest element in the factor vector from H. Problem arise when two or more of the element in H are of similar magnitude. In this case, we can improve the result of clustering from the NMF algorithm, if we could somehow make one of the element of vectors in H significantly larger than the other elements. That is exactly what the CNMF does, it results in even more accurate clustering.

Let us consider one basic scenario, we have data with only 2 attributes, which can be plotted as in 1. If we have to do the feature selection on data then we will leave out Y-value, since all the elements have the same Y-value. If we look at the clusters centroid then will realize that we will get the same set of attributes if feature selection is performed in centroid only. It is because centroid represents more compact representation of the dataset.

IV. Algorithm

Fig. 2. can be summarized as the algorithm given below.

Algorithm 1: Algorithm for feature selection with CNMF

input : A, C
output: A'
1. Perform CNMF on matrix A to obtain H and W.
2. Do Feature selection on W to get W'.
3. Keep only the attributes in W' from A to get feature selected data A'

The procedure requires matrix C in order to perform CNMF of matrix A. After the CNMF is complete, we get

matrix H and matrix W. Feature selection on matrix A is carried out based on the matrix W.

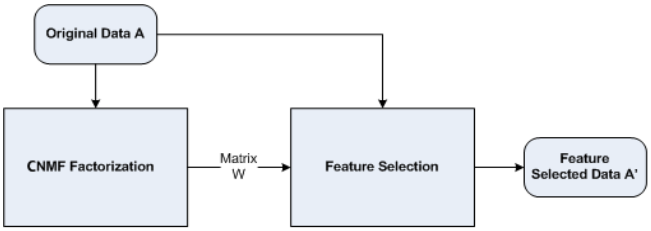


Fig. 2. Procedure for Feature selection using CNMF

A. Time Complexity

In this section, we study the time complexity of our method compared with the traditional method based on CFS. The time complexity of CFS is quite low. It requires $m((n^2-n)/2)$ operations for computing the pairwise feature correlation matrix, where m is the number of instances and n is the initial number of features. The forward selection search method used for feature selection requires $(n^2-n)/2$ operations. In evaluating Equation 11, for a feature subset S containing k features, k additions are required in the numerator (feature-class correlations) and $(k^2-k)/2$ additions are required in the denominator (feature-feature inter-correlations).

$$\begin{aligned} TimeComplexity &= m((n^2-n)/2)+(n^2-n)/2+ \\ &\quad (k^2-k)/2+k \\ &= m*n^2/2-m*n/2-n/2+ \\ &\quad n^2/2+k^2/2+k/2 \\ Simplifying, & \\ &= O(n^2*m) \end{aligned} \tag{13}$$

One simple rule of thumb to set the number of cluster for any dataset is $k \approx \sqrt{n/2}$ with n as the number of objects (data points) [1]. If p be the number of iterations then computational complexity of multiplicative NMF is given by [24] as $p*O(nmk)$. Hence, the computational complexity of our method will be;

$$\begin{aligned} TimeComplexity &= p*O(nmk)+O(n^2*m) \\ &= p*O(nkk)+O(n^2*k) \\ &= p*O(n*n/2)+O(n^2*\sqrt{n/2}) \\ &\approx O(n^2p) \end{aligned} \tag{14}$$

V. Experimental Setup

A. Dataset Used

For our experiments, it was necessary to avoid categorical data, since this research does not deal with categorical data. The domain of dataset we used were drawn from the UCI repository of machine learning databases. These dataset were chosen because of (a) their predominance in literature (b) nature of data being real data.

TABLE I
MRMR based Feature Selection

DataSet	Features	Accuracy(%) CNMF	Accuracy(%) Traditional
LD	15	95.00	81.67
	40	98.33	98.61
	50	99.44	95.56
	70	99.72	100.00
	89	100.00	100.00
WDGD	15	98.62	98.62
	20	99.88	99.90
	25	99.92	99.94
	30	99.94	99.96
	35	100.00	99.98
ISD	10	93.55	80.43
	12	81.30	55.84
	14	97.96	60.26
	16	86.71	100.00
	18	100.00	100.00

- **Libras Dataset (LD)**: The dataset (movement libras) has the size of 360 *91 (including the class).The dataset contains 15 classes of 24 instances each, where each class references to a hand movement type in LIBRAS. The hand movement is represented as a bidimensional curve performed by the hand in a period of time. The curves were obtained from videos of hand movements, with the Libras performance from 4 different people, during 2 sessions.
- **Waveform Database Generator Dataset (WDGS)**: Waveform Data set is a real-valued dataset having 5000 instances and 40 attributes, all of which include noise. The later 19 attributes are all noise attributes with mean 0 and variance 1. The 40 attributes have continuous values between 0 and 6. It has 3 classes of waves. Each class is generated from a combination of 2 of 3 "base" waves. Each instance is generated by add noise (mean 0, variance 1) in each attribute.
- **Image Segmentation Dataset (ISD)**: Image Segmentation dataset is a real dataset with 2310 instances in a 19-dimensional attribute space. The four attributes are sepal length, sepal width, petal length and petal width. The dataset contains 7 classes: brickface, sky, foliage, cement, window, path, grass. It has 30 instances per class for training data and 300 instances per class for

TABLE II
Correlation based Feature Selection (CFS)

Dataset	Features	Accuracy(%) CNMF	Accuracy(%) Traditional
LD	50	99.44	99.17
	60	100.00	98.89
	70	100.00	99.44
	80	100.00	99.72
WDGD	20	97.96	99.16
	25	98.72	99.16
	30	99.74	99.68
	35	99.76	99.90
ISD	9	65.06	48.13
	12	57.79	77.74
	15	71.95	83.33
	18	99.65	100.00

TABLE III
COMPUTATIONAL TIME(SEC) FOR FEATURE SELECTION USING MRMR

Dataset	Features	Time Using CNMF		Time for mRMR
		Factorization	FS	
LD	40	0.109	0.374	0.530
	55	0.234	0.437	0.530
	70	0.125	0.484	0.842
	85	0.686	0.577	0.624
WDGD	20	0.593	0.109	0.219
	25	1.357	0.094	0.250
	30	0.343	0.109	0.343
	35	0.733	0.156	0.452
ISD	12	0.749	0.016	0.156
	14	0.827	0.047	0.203
	16	1.451	0.062	0.390
	18	0.218	0.062	0.406

test data.

B. Experimental Results

The paper presents the experiments that we conducted and the results achieved in the following sections. Two types of observations were made, first was to observe the clustering accuracy while the second one dealt with the efficiency in terms of execution time. Comparison was made between the traditional methods and our method. First two experiments dealt with non-discretized data, while the third experiment was with discretization and FS on discretized data.

1) Experiment 1: Clustering Accuracy: Our first experiment dealt with observing the utility of the feature selected datasets using our method compared to the traditional method. We measured the accuracy of each of the method by comparing the k-means result of feature selected dataset with that of the original data. From Table I, it can be observed that the accuracy of the two methods are comparable with one falling behind the other method occasionally. From Table II, the results indicates the new method produces results comparable with the traditional method.

2) Experiment 2: Time for computing feature selected data without discretization: As the size of the matrix on which feature selection is done in our method is small than the original dataset, it was very intuitive to test the execution time for our algorithm compared to the traditional methods. From Table III, we can see that our method may or may not run faster than the traditional methods if we take into account the matrix factorization. Only considering the feature selection process (excluding the time taken for factorization) our method is much faster. Our method can still be useful in cases where we need to do the feature selection more than once in the same set of factorized matrix.

TABLE IV
COMPUTATIONAL TIME(SEC) FOR FEATURE SELECTION USING CFS

Dataset	Features	Time for Method Using CNMF		Time for CFS
		Factorization	FS	
LD	10	0.22	0.037	0.64
	30	0.20	0.022	1.3
	50	0.25	0.046	0.98
	70	0.09	0.128	1.63
WDGD	15	0.51	0.008	2.54
	20	0.57	0.009	3.06
	25	0.58	0.003	6.22
	30	1.67	0.001	5.34
	35	1.42	0.003	4.90
ISD	9	0.31	0.070	0.09
	12	0.27	0.001	0.11
	15	0.22	0.002	0.13
	18	0.20	0.007	0.17

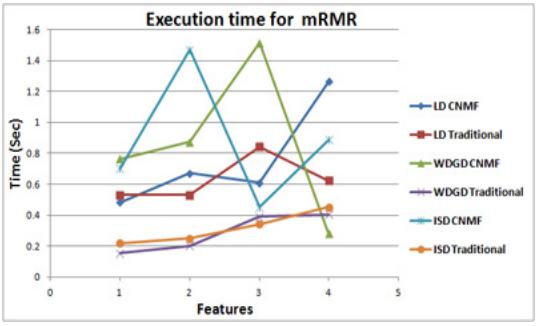


Fig. 3. mRMR on non-discretized data

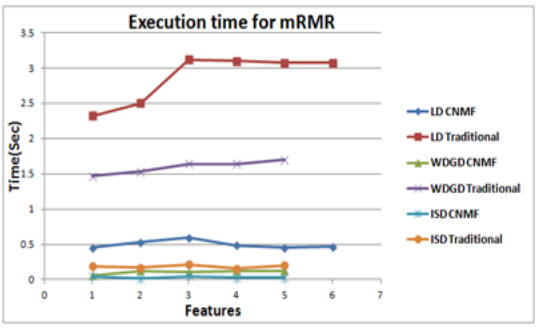


Fig. 4. mRMR on discretized data

From Fig. 3., we cannot really say which method executes faster, it depends on various things: dataset, initialization matrix for CNMF, and the number of iteration that needs to be performed.
From Table IV, similar observations were made for CFS.

3) Experiment 3: Time for computing feature selected data with discretization: In our third experiment, we compare the execution time of the feature selection process with discretization. Table V shows the execution time of methods for discretization. We can clearly see that you method takes lot lesser time to perform the discretization than using traditional method.

Table VI shows that k-means using our method definitely has better accuracy. This does not only mean that our method is faster but also the results are better than using the traditional method.

Fig. 4. is the graph of execution time for mRMR on discretized data. Although the feature selection in itself may or may not be faster than the traditional method but we definitely have edge while discretizing data.

TABLE V
COMPUTATIONAL TIME(SEC) FOR DISCRETIZATION

Dataset	Discretization Algorithms					
	CAIM		CACC		Ameva	
	CNMF	Traditional	CNMF	Traditional	CNMF	Traditional
LD	28	532	13.76	4254	10.9	474.05
WDGD	0.16	808.38	0.855	33358	0.17	876.90
ISD	9.31	285.04	0.35	5.74	0.39	290

TABLE VI
COMPUTATIONAL TIME (SEC) FOR FEATURE SELECTION ON DISCRETIZED DATA USING MRMR

Dataset	Features	Time Using Kmeans			
		with NMF		without NMF	
		Accuracy	Time	Accuracy	Time
LD	60	81.94	0.530	68.61	2.50
	65	84.17	0.593	68.61	3.12
	70	87.50	0.484	67.50	3.10
	75	89.44	0.452	70.56	3.07
	80	91.94	0.468	78.06	3.07
WDGD	20	98.78	0.062	99.86	1.467
	24	99.90	0.125	99.90	1.529
	28	99.90	0.109	99.90	1.638
	32	99.94	0.125	99.96	1.638
	36	99.96	0.125	100.00	1.700
ISD	6	53.80	0.047	80.28	0.187
	9	100.00	0.016	80.28	0.171
	12	100.00	0.047	87.64	0.218
	15	100.00	0.031	89.02	0.156
	18	100.00	0.031	100.00	0.203

VI. CONCLUDING REMARKS

We proposed a novel method for feature selection that uses constraint based nonnegative matrix factorization. The paper experimentally shows results from the two methods in terms of clustering accuracy and execution time. From the experiments we can conclude following things;

1) NMF based feature selection works equally as good as the traditional methods in terms of accuracy.
2) NMF based methods can be significantly efficient compared to traditional methods specially when discretization is required.

REFERENCES

[1] Kanti Mardia et al. "Multivariate Analysis,"*Academic Press*, 1979.

[2] Hanchuan Peng, Fuhui Long, and Chris Ding, "Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy,"*IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 8, pp.1226-1238, 2005.

[3] Jie Wang, Jun Zhang, Lian Liu, and Dianwei Han, "Simultaneous data and pattern hiding in unsupervised learning,"*The 7th IEEE International Conference on Data Mining - Workshops(ICDMW07)*, pages 729-734, Omaha, NE, USA, October 2007.

[4] Jie Wang, Weijun Zhong, and Jun Zhang, "NNMF-based factorization techniques for high-accuracy privacy protection on non-negative-valued datasets,"*2006 IEEE Conference of Data Mining, International Workshop on Privacy Aspects of Data Mining*, pp. 513-517. IEEE Computer Society, 2006.

[5] Y. Gao and G. Church, "Improving molecular cancer class discovery through sparse nonnegative matrix factorization,"*Bioinformatics*, 21(21):3970-3975, 2005.

[6] P. Langley and S. Sage, "Oblivious decision trees and abstract cases,"*In Working Notes of the AAAI-94 Workshop on Case-Based Reasoning*, Seattle, W.A, 1994. AAAI Press.

[7] P. Langley and S. Sage, "Scaling to domains with irrelevant features,"*In R. Greiner, editor, Computational Learning Theory and Natural Learning Systems*, volume 4. MIT Press, 1994.

[8] P. Langley and S. Sage, "Induction of selective Bayesian classifiers,"*In Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, Seattle, W.A, 1994. Morgan Kaufmann.

[9] D.W. Aha, D. Kibler, and M. K. Albert, "Instance based learning algorithms,"*Machine Learning*, 6:37-66, 1991.

[10] Nirmal Thapa, Lian Liu, Pengpeng Lin, Jie Wang, and Jun Zhang, "Constrained Non-negative Matrix Factorization for Data Privacy,"*Proceedings of the International Conference on Data Mining (DMIN)*, Las Vegas, Nevada, July 17-21, 2011.

[11] Li, H., Adal, C., Wang, W., Emge, D., and Cichocki, A., "Non-negative matrix factorization with orthogonality constraints and its application to raman spectroscopy,"*The Journal of VLSI Signal Processing*, 48,83-97 (2007).

[12] Patrik O. Hoyer, "Non-negative matrix factorization with sparseness constraints,"*Journal of Machine Learning Research*, 5:1457-1469, 2004.

[13] H. Liu and H. Motoda,"Feature Extraction, Construction and Selection: A Data Mining Perspective,"*Boston: Kluwer Academic*, 1998, second printing, 2001.

[14] H. Liu and H. Motoda,"Feature Selection for Knowledge Discovery and Data Mining,"*Boston: Kluwer Academic*, 1998.

[15] A.L. Blum and P. Langley, "Selection of Relevant Features and Examples in Machine Learning,"*Artificial Intelligence*, vol. 97, pp. 245-271, 1997.

[16] H. Almuallim and T.G. Dietterich, "Learning Boolean Concepts in the Presence of Many Irrelevant Features,"*Artificial Intelligence*, vol. 69, nos. 1-2, pp. 279-305, Nov. 1994.

[17] W. Xu, X. Liu, and Y. Gong, " Document clustering based on non-negative matrix factorization,"*In Proc. SIGIR*, pp.267273, 2003.

[18] U.M. Fayyad and K.B. Irani, "The Attribute Selection Problem in Decision Tree Generation,"*Proc. AAAI-92, Ninth Intl Conf. Artificial Intelligence*, pp. 104-110. AAAI Press/The MIT Press, 1992.

[19] K. Kira and L.A. Rendell, "The Feature Selection Problem: Traditional Methods and a New Algorithm," *Proc. AAAI-92, Ninth Intl Conf. Artificial Intelligence* , pp. 129-134. AAAI Press/The MIT Press, 1992.

[20] H. Liu and R. Setiono, "A Probabilistic Approach to Feature Selection A Filter Solution," *Proc. 13th Intl Conf. Machine Learning*, pp. 319-327, 1996.

[21] Kurgan, L.A., Cios, K.J., "CAIM Discretization Algorithm,"*IEEE Transactions on Knowledge and Data Engineering*, 16:2 (2004) 145-153.

[22] Tsai, C., Lee, C., Yang, W. "A discretization algorithm based on class-attribute contingency coefficient,"*Information Sciences* 178, 714-731 (February 2008)

[23] L. Gonzalez-Abril, FJ Cuberos, F. Velasco, and JA Ortega. "Ameva: An autonomous discretization algorithm," *Expert Systems with Applications*, 36(3):5327-5332, 2009.

[24] Lin, C.-J., 2005b. "Projected gradient methods for non-negative matrix factorization,"*Tech. Rep. Information and Support Services Technical Report ISSTECH-95-013*, Department of Computer Science, National Taiwan University.

[25] E. E. Ghiselli, "Theory of Psychological Measurement,"*McGrawHill*, New York, 1964.

[26] R. M. Hogarth, "Methods for aggregating opinions," *In H. Jungermann and G. de Zeeuw, editors, Decision Making and Change in Human Affairs. D. Reidel Publishing*, Dordrecht-Holland, 1977.

[27] R. B. Zajonic,"A note on group judgements and group size,"*Human Relations*,15:177-180, 1962.

[28] Caruana, R.,and Freitag, D, "Greedy attribute selection,"*In Cohen, W., W., and Hirsh, H., eds., Machine Learning Proceedings of the Eleventh International Conference*, Morgan Kaufmann,

Generating Rules to Increase Production Using Decision Tree

Keivan Ghoseiri, Hassan Gholami Mazinan, Mahyar Hoseinzadeh, Maziar Davoodi, Erfan khaji

***Abstract*— Production development is a crucial subject which has been deeply investigated with various methods such as heuristic approaches, data mining, production plan optimization and etc. This paper extracts proper rules and removes weak rules for production planning using decision trees and compares results with expert judgmental rules. Evaluating the results verifies that production planning based on the extracted rules from decision trees incur a significant amount of improvement in production development which is a good indication for efficiency of data mining methods in production planning.**

I. Introduction

Data mining in various forms is becoming a major component of business operations. Even though data mining has been successful in becoming a major component of various business processes as well as in transferring innovations from academic research into the business world, the gap between the problems that the research community works on and real world is still significant [1]. Almost every business process today involves some forms of data mining. Customer Relationship Management, Supply Chain Optimization, Demand Forecasting, Assortment Optimization, Business Intelligence, and Knowledge Management are just some examples of business functions that have been impacted by the data mining techniques [2].

Designing an effective production plan, achieving the production plan and then in-creasing the production according to the present conditions are some of the big-gest challenges in the large organizations. There are different approaches for designing production plans. Some approaches try to design a production plan by mathematical models [3], Meta heuristic algorithms [4], simulation [5] and analytical and heuristics methods [6]. Otherwise some approaches instead of using the new method, tried to evaluate existing conditions and relation between them to improve the amount of production. Chen & Cochran [7] and Chen et al. [8] studied about the Driving Daily Production Plans by Effectiveness of Manufacturing Rules. In the papers [9] and [10] are proposed multi-objective daily production plans for complex manufacturing facilities.

K. Ghoseiri is faculty research assistant, University of Maryland, College Park, USA (corresponding author to provide phone: 202-706-4284; e-mail: kghoseir@umd.edu).

H. Gholami Mazinan work in SAIPA KASHAN (+983614911668), and is graduate student in School of Railway Engineering, IUST, Iran (hassan_mazinan@rail.iust.ac.ir).

M. Hoseinzadeh work in SAIPA KASHAN (+983614911671), he is graduate student in Industrial Engineering department, Tarbiat Modares University, Iran (h.mahyar@gmail.com).

M. Davoodi work in SAIPA KASHAN (+983614911666), and is graduate student in Industrial Engineering department, IUST, Iran (davoodi.samyar@gmail.com).

E. Khaji is graduate student in Department of Physics, Goteborg University, 41296 Gothenburg, Sweden. (erfankhaji@gmail.com).

II. Problem Description

Some problems and constraints in production plan are result of the existing inappropriate rules or lack of appropriate rules. Figure 1 shows flow of process and operations in an original equipment manufacturer (OEM).

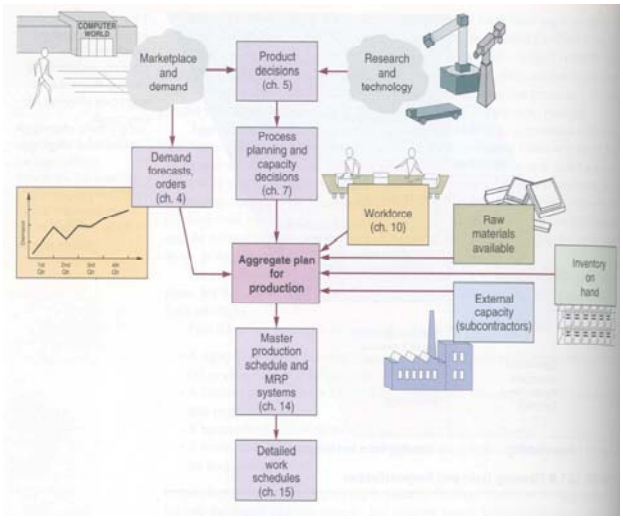


Fig.1. Production-related Decision Making in Large Corporations [11]

Saipa Kashan Company has been established in the 14th kilometer of Kashan-Ardestan road, Isfahan province in 2008 on area of 420 Ha. This company has been producing 3 types of cars, Tiba (two classes), X111(three classes), and X132 (three classes) by about 3500 personnel. This company includes two warehouses (including warehouse of assembly parts and warehouse of body parts), body production line, color production line and assembly line (which are working in 2 platforms).

This research will discuss the reasons of non-fulfillment of production plan in this company. To reach this goal, we have to investigate all factors that are affected in the amount of production or those could cause the stop in the production plan. Also a field investigation has been carried out to find out which of these factors are more repetitive and important in stopping of production line directly or indirectly. Among lots of items, 14 items have been chosen as the most affective factors in the production line stops. These factors are:

TABLE 1
THE MOST AFFECTIVE FACTORS IN THE PRODUCTION LINE STOPS

factor	description
Date	This parameter shows the date. In this research the data has been gathered from 1/1/2011 in 1/1/2012 (1390/10/11 in 1389/10/11 in Persian) period of time in each working day separately.
Shift	Since this company has got 3 different working shifts, this parameter could show us whether if different shifts affect production trend or not.
P_day	This one could determine whether or not different working days in a week affect occurrence of production plan and how important and repetitive this factor is.
A-per-L	This item is the number of absent personnel of logistic department and investigates this factor in recoding of stops.
Ab-per-P	This considers the number of absent personnel of production line and investigates it in the amount of production and stops statistic.
PRS	This parameter is the amount of production in recent shift with respect to last one (production rate on the two last shifts).
Day	This parameter considers the date of investigation which can be in first, second or third decade of month. Since the fraction of transferred goods in these three decades is not the same, e.g. the fraction of transferred parts in last days of month is more, so that it can play a significant role.
Climate	This is one of parameters that have a very important role on analysis and the total results as well.
Crisis	It mentions the parts that are critical while starting a shift. In this definition, critical parts are those which are less than 50 in the production line or in the warehouse in the starting of a shift. This shortage could affect fulfillment of production plan.

III. COMPUTATIONAL RESULTS

The recent paper has got 3 purposes:

1) Analysis of collected data and identifying effective factors in production per shift.

2) The effect and importance of factors on production per shift.

3) Forecasting the condition of production per shift with regards to accessible values in first of a shift.

Therefore, the decision tree method is used since decision tree method makes possible to all of our accessible aims.

The classic CART algorithm was propos by Breiman et al. [12]. CART is a recursive partitioning method, builds classification and regression trees for predicting continuous dependent variables (regression) and categorical predictor variables (classification).

The QUEST algorithm (Quick, Unbiased, Efficient Statistical Trees), is also presented in the context of the Classification Trees Analysis facilities, and much of the following discussion presents the same information, in only a slightly different con-text. Another, similar type of tree building algorithm is CHAID (Chi-square Automatic Interaction Detector) [13].

The CHAID is one of the oldest tree classification methods originally proposed by Kass (1980) [14]. According to Ripley (1996) [15], the CHAID algorithm is a descendent of THAID developed by Morgan and Messenger, (1973) [16]. CHAID will "build" non-binary trees (i.e., trees where more than two branches can attach to a single root or node), based on a relatively simple algorithm that is particularly well suited for the analysis of larger datasets. Also, because the CHAID algorithm often effectively yield many multi-way frequency tables (e.g., when classifying a categorical response variable with many categories, based on categorical predictors with many classes), it has been particularly popular in marketing research, in the context of market segmentation studies. CHAID is a recursive partitioning method.

Both CHAID and C&RT techniques will construct trees, where each (non-terminal) node identifies a split condition, to yield optimum prediction (of continuous dependent or response variables) or classification (for categorical dependent or response variables). Hence, both types of algorithm can be applied to analyze regression-type problems or classification-type.

The results of CHAID, CART and QUEST tree is shown in the following. Fig 1, 2 and 3 illustrate the plot of importance of variables for three methods CHAID, CART and QUEST.

All of this figures identified the V_col, V_Pro and Crisis variables are the most important variable. Moreover, fig 4 and 5, show the trees resulting the CART and QUEST methods.

The table below, shows the importance of variables in 3 decision tree methods

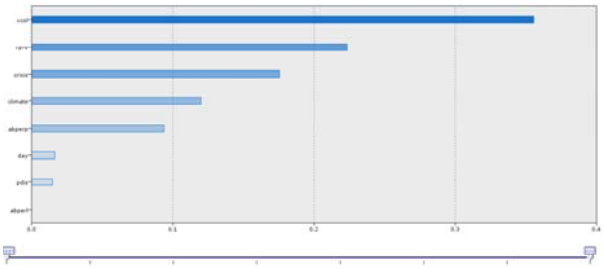


Fig.2. Important factor based on CHAID

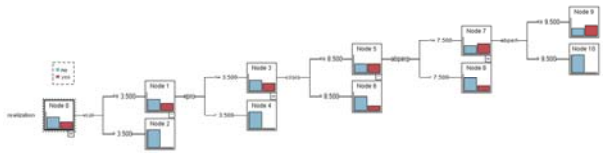


Fig.3. Important factor based on CART

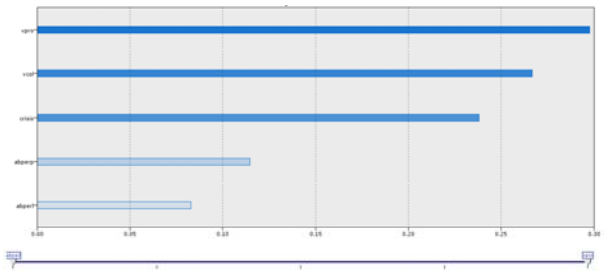


Fig.4.Important factor based on CART

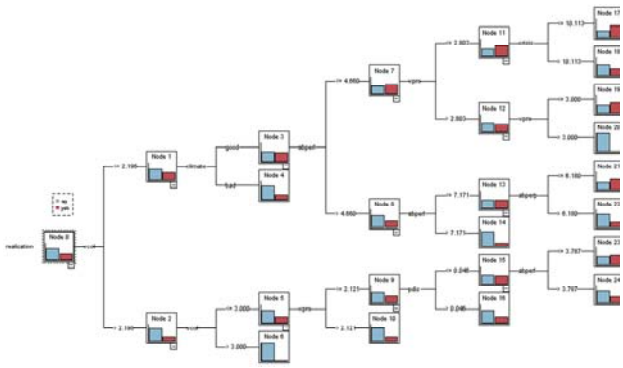


Fig.5. Important factor based on QUEST

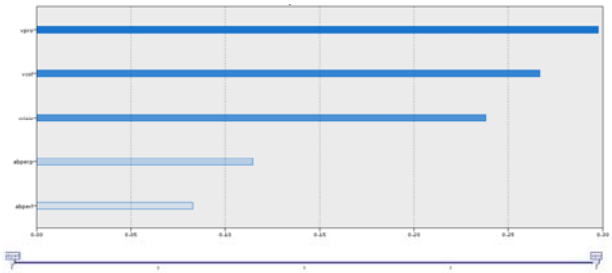


Fig.6. Important factor based on QUEST

All of this figures identified the V_col, V_Pro and Crisis variables are the most important variable. Moreover, fig 4 and 5, show the trees resulting the CART and QUEST methods.

The table below, shows the importance of variables in 3 decision tree methods

TABLE 2
THE IMPORTANCE OF VARIABLES IN 3 DECISION TREE METHODS

	CHAID	CART	QUEST	Total
P_Dis	0.015	0	0.021	0.036
Ab_Per_L	0	0.083	0.068	0.151
Ab_Per_P	0.094	0.115	0.003	0.212
Climate	0.12	0	0.167	0.287
Crisis	0.176	0.238	0.028	0.442
V_Pro	0.224	0.298	0.209	0.731
V_Col	0.365	0. 267	0.47	0.835
Day	0.017	0	0	0.017

These three methods show that variety of color is the most important factor in production. The results of decision tree algorithms show that in all shifts in the production increasing in which there are three different colors, they will encounter with de-fulfillment of production plan. Analyses of this note helps the expert of production plan to either presents a scheduling with at most three different colors per shift.

According to logistic and production experts, producing more than three colors diversity per shifts, lead to some problems:

1) Inability to supply colored parts (for example bumper) with more than three colors diversity per shift by supply management department.

2) Inability to product colored parts with more than three colors diversity per shifts by supply chain.

3) Inability to feeding colored parts with more than three colors diversity per shift by feeding lines management department.

4) The lack of enough space besides production line to storage colored parts with three colors diversity per shift.

5) Inability workstations to produce colored parts with more than three colors diversity per shift.

Understanding and solving the available problems helps to achieve a better production plan. After colors diversity, another important factor to increase the production is variety of product. Based on the decision tree algorithms, if varieties of products are more than three types per shift, Production plan will not be achieved. The results of decision tree algorithms show whenever the numbers of critical parts in the first of a shift are more than 8; Production plan will not be achieved. The planning department experts can re-schedule production plan and put those products in the program that make less number of critical parts. The supply management department experts can use other suppliers to supply critical items or transmit them to the factory with higher cost.

In addition to items mentioned above, there are some rules extracted from trees that are as follows:

• If colors diversity is more than 3, the production program will be stopped.

• If colors diversity is less than 4 in 2 shifts, but the variety of products is more than 3, we will not have the scoping production.

If colors diversity is less than 3 but climate is unfavorable or harsh conditions, the production program won't be stopped by 75%.

In this paper, based on the results of decision tree algorithms, proposed some solutions to increase production. Table 3 shows some important solutions that they have more effects in this purpose.

TABLE 3
THE IMPORTANCE OF VARIABLES IN 3 DECISION TREE METHODS

No.	Solution	Description
1	Two labors employed share between in the logistic and production units	Based on absent persons in logistic or production units, Managers can be decide about the position of these labors
2	Determining the maximum colors diversity per shift	Maximum color diversity equal 3
3	Balance between production plan in maximum color diversity and maximum production variety per shift in unfavorable climate conditions	For example in unfavorable climate conditions: If the colors diversity equal 3, the maximum production type must be 2 (If the colors diversity equal 1, the maximum production type must be 3)
4	Balance between colors diversity and production type	Total variation of production type and colors diversity must be 6.

After experimental implementation of these solutions in the Saipa Kashan for 3 month, the percentage of production plan increased about 8%. The results of this project include the following:

- Decrease Variable costs (including costs of logistic, production lines stop, reducing waste of materials by expiration date ...)
- Increased profit from higher productivity
- Increase customer satisfaction
- Move towards a quality management approach and customers satisfaction after fulfillment production plan.

IV. SUMMARY AND CONCLUSIONS

This paper discusses the different approaches in production planning. One approach is identification of the existing inappropriate rules and generation of appropriate rules. In Saipa Kashan the production planning expert use the existing rules at the first of shift to increase possibility of fulfillment of production plan. In the presented work, the data are analyzed using decision tree to identify importance factors in the production. The achieved rules identify the issue of unscheduled failures in the production program which makes possible for the experts to investigate the most significant problems and find a solution for them. The proposed methods help us to extract the hidden science of achieved data. Moreover, decision tree is so suitable for fore-casting since it's simple in use and applicable. Also, the resulting rules of decision tree methods are accessible and understandable.

REFERENCES

[1] S. Gunasekaran, C. Chandrasekaran, "A survey on automobile industries using data mining techniques," International Journal of Science and Advanced Technology, Volume 1 No 4, 2011, pp. 30-35.

[2] J. Han, M. Kamber, "Data Mining Concepts and Techniques". 2nd Edition, the Morgan Kaufmann Publishers, ISBN 1-55860-901-6, 2006.

[3] X. Qu, J. A. Stuart Williams, "An analytical model for reverse automotive production planning and pricing," European Journal of Operational Research, Vol. 190, Issue 3, 2008, pp. 756–767.

[4] D. F. Zhu, H. Wang, S. B. Wang, J. X. Xu, "Hybrid Genetic Algorithm-Based Production Planning for Steel-Making and Continuous Casting Process," Advanced Materials Research, Vols. 383 - 390, 2011, pp. 1677-1683.

[5] N. B. Kacar, D.F. Irdem, R. Uzsoy, "An Experimental Comparison of Production Planning Using Clearing Functions and Iterative Linear Programming-Simulation Algorithms," Semiconductor Manufacturing, IEEE Transactions on, Vol. 25, 2012, pp. 104-117.

[6] M. G. Gnoni, R. Iavagnilio, G. Mossa, G. Mummolo, A. Di Leva, "Production planning of a multi-site manufacturing system by hybrid modeling: A case study from the automotive industry", Supply Chain Management, Vol. 85, 2003, pp. 251–262.

[7] H. N. Chen, J. K. Cochran, "Effectiveness of manufacturing rules on driving daily production plans," Journal of Manufacturing Systems, Vol. 24, Issue 4, 2005, pp. 339–351.

[8] H.N. Chen, J.K. Cochran, R.M. Dabbas, "Using manufacturing rules to implement daily production plans," Modeling and Analysis of Semiconductor Mfg. Conf., 2002, pp175-181.

[9] J. K. Cochran, H. N. Chen, "Generating daily production plans for complex manufacturing facilities using multi-objective genetic algorithms," International Journal of Production Research, Vol. 40, Issue 16, 2002.

[10] H.N. Chen, R.M. Dabbas, J.K. Cochrau, "Optimizing multi-objective production and setting daily production goals for a wafer fabrication facility," Semiconductor Mfg. Operational Modeling and Simulation Symp., 2001, pp. 63-68.

[11] J.H. Heizer, B. Render, "Operations Management," Prentice Hall, 0136119417, 9780136119418, 2010.

[12] L. Breiman, J. Friedman, R. Olshen, C. Stone, "Classification and Regression Trees," Wadsworth, Belmont, CA, 1984.

[13] W.Y. Loh, Y.S. Shih, "Split selection methods for classification trees," Statistica Sinica, vol. 7, 1997, pp. 815-840.

[14] G.V. Kass, "An Exploratory Technique for Investigating Large Quantities of Categorical Data," Applied Statistics, Vol. 29, No. 2, 1980, pp. 119–127.

[15] B.D. Ripley, "Pattern recognition and neural networks," Cambridge: Cambridge University Press, 1996.

[16] J.N. Morgan, R.C. Messenger, "THAID: A sequential analysis program for the analysis of nominal scale dependent variables," Institute of Social Research, University of Michigan, Ann Arbor. Technical report, 1973.

Classification and Regression Trees for Handling Missing Values in a CMBD to reduce malware in an Information System.

Gustavo A Valencia-Zapata, Juan C Salazar-Uribe, Ph.D.
Escuela de Estadística, Universidad Nacional de Colombia-Sede Medellín
gavalenciaz@unal.edu.co, jcsalaza@unal.edu.co

***Abstract*— In this paper we propose a Classification and Regression Trees model (CART) for handling missing values in a Configuration Management Database (CMDB). Once the information is completed a statistical model to dose antivirus scans inside an information system (IS) in banking sector is implemented. Since about 18.22% of the extracted information from the CMBD was incomplete. As a consequence we propose a data mining modeling strategy to impute this missing information. Finally, we illustrated both this imputation methodology and the statistical dosage model using real data from an IS.**

KeyWords: CART, Missing Values, Data Mining, Banking Sector, Malware.

I. Introduction

THIS research is intended to be a resource to improve the information security levels in banking sector (IS). The research question is: How malware incidence can be decreased in an IS? As in human epidemiologic context is necessary to apply treatments (medicine, vaccines, therapies, etc.), on a computer environment would be the application of scanning. Based on that, we can reformulate the above question as: How antivirus scans (medical tests) can be dosed, in our population (computer network), for the reduction of malware (disease) incidence in banking IS? Different approaches have been made in modeling 'disease' for Information Technology (IT) environment, using some analogies with the epidemiologic context [1]. In this sense, an exponential growth of malware has been observed in the last decades, as well as, the outlook and limitations of epidemiological concepts for malware prevention [1]. Also, malware spreading and measurements models had been elaborated [2], [3], [4]. Similarly, simulation of the networking topology influence in malware problems had been discussed by [5], [6]. Finally, epidemiological methodologies are used to estimate growth and propagation of worms in a network [7].

In this paper the first stages to build the model are: information extraction (IE), handling missing values, and statistics analysis. The main information source is the bank antivirus software. Secondary information sources are: web filtering, Human Capital/Resource Management-HCM (company employees), and CMDB. Physiological computer information is provided by CMDB such as: brand, operating system, processor type, random access memory (RAM), and so on. CMDB parameters are showed in Table I. Using data mining software (through Open Database Connectivity, ODBC) we collect information about malware attacks over a period of eleven weeks. Then, secondary sources information is added as can be seen in Figure 1. Around 18.22% of CMDB data (infected computers) are missing values. Classification and Regression Trees (CART) are used for handling missing values (imputation) to avoid losing valuable information. This research used nonparametric Statistical tests for checking the quality of the imputed data. Moreover, statistical analysis is conducted to select variables that will be included into the antivirus scanning dosage model

II. Information Collection Stages

A. Antivirus Software

In this stage, amount and type of malware per computer are identified. Information over a period of eleven weeks about 8476 computers was collected. Antivirus software reports the active user account when malware was detected and some other technical information of the computer.

TABLE I
CMDB Parameters

Variable	Meaning/value	Type	Unit
Class	Laptop, CPU or server	Nominal	NA
Brand	Computer brand	Nominal	NA
Computer_Age	Operating time	Scale	Week
Processor_Type	Type of computer processor	Nominal	NA
Processor_Clock	The speed of a computer processor	Scale	GHz
Processors	Number of processors	Integer	Count
Memory (RAM)	Memory size	Scale	GB
Operation_System	Operation System (OS)	Nominal	NA
Service_Pack	Updates to a OS	Nominal	NA
Hard_Disk	Hard disk size	Scale	GB

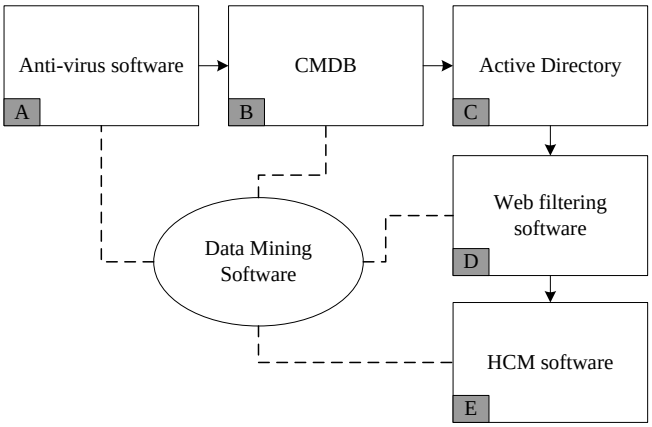


Fig. 1. Informationcollection stages.

TABLE II
CMDB DATA QUALITY

Variable	Percent Complete	Valid Records	Missing Records
Class	83.483	7076	1400
Brand	83.483	7076	1400
Computer_Age	81.784	6932	1544
Processor_Type	99.493	8433	43
Processor_Clock	99.493	8433	43
Processors	99.493	8433	43
Memory (RAM)	99.493	8433	43
Operation_System	99.493	8433	43
Service_Pack	99.457	8430	46
Hard_Disk	99.493	8433	43

B. CMDB

Here, technical information about computers is identified as well as its relationship with user´s account. Table II shows CMDB variables, percent completed valid and missing records. This information is used to assess the influence of these variables over detected malware levels.

C. Active Directory

At this point, the user's privileges are identified. For example: adding a user to the Local Administrator Group or some user accounts are allowed to use USB devices. In this way, privileges acquaintance is important to establish whether or not these variables have some influence over malware levels.

D. Web Filtering Software

At this step, user account is related to web surfing, identifying variables such as: number and type of blocked websites, web surfing time, etc. The main purpose of studying this association is to establish if web surfing behavior has influence on malware levels.

E. HCM Software

In this stage, employee information is identified. Collected information such as position and work area is used to assess the influence of these variables over detected malware levels.

III. CMDB DATA IMPUTATION

A. Classification and Regression Trees, CART

CART model is explained in detail in [8]. The classification and regression trees (CART) method was suggested by Breiman et al. [8]. According to Breiman, the decision trees produced by CART are strictly binary, containing exactly two branches for each decision node. CART recursively partitions the records with similar values for the target attribute. The CART algorithm grows by conducting for each decision node, an exhaustive search of all available variables and all possible splitting values, selecting the optimal split according to the following criteria [9].

Let $\varphi(s|t)$ be a measure of the "goodness" of a candidate split s at node t, where

$$\varphi(s|t) = 2P_L P_R \sum_{j=1}^{\#\,classes} |P(j|t_L) - P(j|t_R)| \qquad (1)$$

Split parameters are defined in Table III. One of the major contributions of CART was to include a fully automated and effective mechanism for handling missing values [10]. Decision trees require a missing value-handling mechanism at three levels: (a) during splitter evaluation, (b) when moving the training data through a node, and (c) when moving test data through a node for final class assignment [11].

TABLE III
SPLIT PARAMETERS

Parameter	Meaning
t_L	Left child node of node t
t_R	Right child node of node t
P_L	$\frac{\text{Number of records at } t_L}{\text{Number of records in training set}}$
P_R	$\frac{\text{Number of records at } t_R}{\text{Number of records in training set}}$
$P(j\|t_L)$	$\frac{\text{Number of class j records at } t_L}{\text{Number of records at } t}$
$P(j\|t_R)$	$\frac{\text{Nnumber of class j records at } t_R}{\text{Number of records at } t}$

According to [11], regarding (a), the later versions of CART (the one we use) offers a family of penalties that reduce the improvement measure to reflect the degree of missingness. (For example, if a variable is missing in 20% of the records in a node then its improvement score for that node might be reduced by 20%, or alternatively by half of 20%, and so on.) For (b) and (c), the CART mechanism discovers "surrogate" or substitute splitters for every node of the tree, whether missing values occur in the training data or not. The surrogates are thus available, should a tree trained on complete data be applied to new data that includes missing values.

B. Handling Missing Values through Classification and Regression Trees

Ten variables were imputed (Table II), that is, ten CART were used, a CART for each variable, which together make up a classification and regression forest. The imputation was made using PASW® Modeler (a data mining software). Model training was made with complete data and then, this trained model was applied to missing values. Table VI and Table VII show a nonparametric statistics test called *McNemar Test for Significance of Changes* [12], which evaluated model prediction by using complete data. In this case *E_Class* is the imputed value and *Class* is the real value. For instance, 5013 (99.6%) computers with *Class* equal to CPU were classified correctly by CART, and 2002 (99.3%) computers with *Class* equal to Laptop were classified correctly by the same CART.

The formulated hypotheses for McNemar test (2-sided) were:

$Null\ hypothesis, H_0 =$ Class is not changed after imputation.

$Alternative\ hypothesis, H_1 =$ Class was changed after imputation.

According to this analysis we cannot reject the null hypothesis, that is, CART doesn't change *Class* values after imputation (p-value=0.396). Figure 2 Shows a CART model for *Class* variable that was built using the software PASW® Modeler.

Spearman's Test (a nonparametric statistic) was used to explore the correlation between real and predicted values for continuous variables. On the other hand, McNemar´s test was used to assess the performance of the CART Model with complete data. In this case, *E_Computer_Age* is the imputed value whereas *Computer_Age* is the real value.

TABLA IV
CONTINGENCY TABLE CLASS

		E_Class		Total
		CPU	Laptop	
Class	CPU	5.013	20	5.033
	Laptop	14	2002	2.016
Total		5027	2022	7049

TABLA V
CHI-SQUARE TEST

	Value	Exact Sig. (Two-sided)
McNemar Test	1.058	0.392
Nº Valid Cases	7049	

Use binomial distribution

Figure 2 shows the scatterplot for these variables. We can observe a linear trend between the imputed and real values ($r^2 = 0.9085$). So we fitted a possible simple linear regression. Nevertheless, assumptions validation of the linear model is not the aim this article. Spearman´s test was used to test independence. The formulated hypotheses for this test were:

$Null\ hypothesis, H_0 =$ X_i and Y_i are mutually independents.

$Alternative\ hypothesis, H_1 =$ X_i and Y_i are not mutually independent.

According to the results from this test X_i and Y_i are not mutually independent (p-value<0.0001). In particular, X_i are *Computer_Age* data and Y_i are *E_Computer_Age* data. As a result, CART model for Computer*_Age* data imputation is reliable. Figure 3 Shows a CART model for *Class* variable that was built using the software PASW® Modeler.

In particular, Table VI shows the variables for the computer number 0022. We can identify three out of ten variables with missing values. Node 0 indicates that CPU category has the higher probability (0.7) to be selected if a random imputation is conducted. On the other hand, Laptop variable has a smaller probability (0.28) than the first one and the Server variable has null probability (0.0). As can be seen, we should slide through CART's branches according to the values of the variables shown in Table VI.

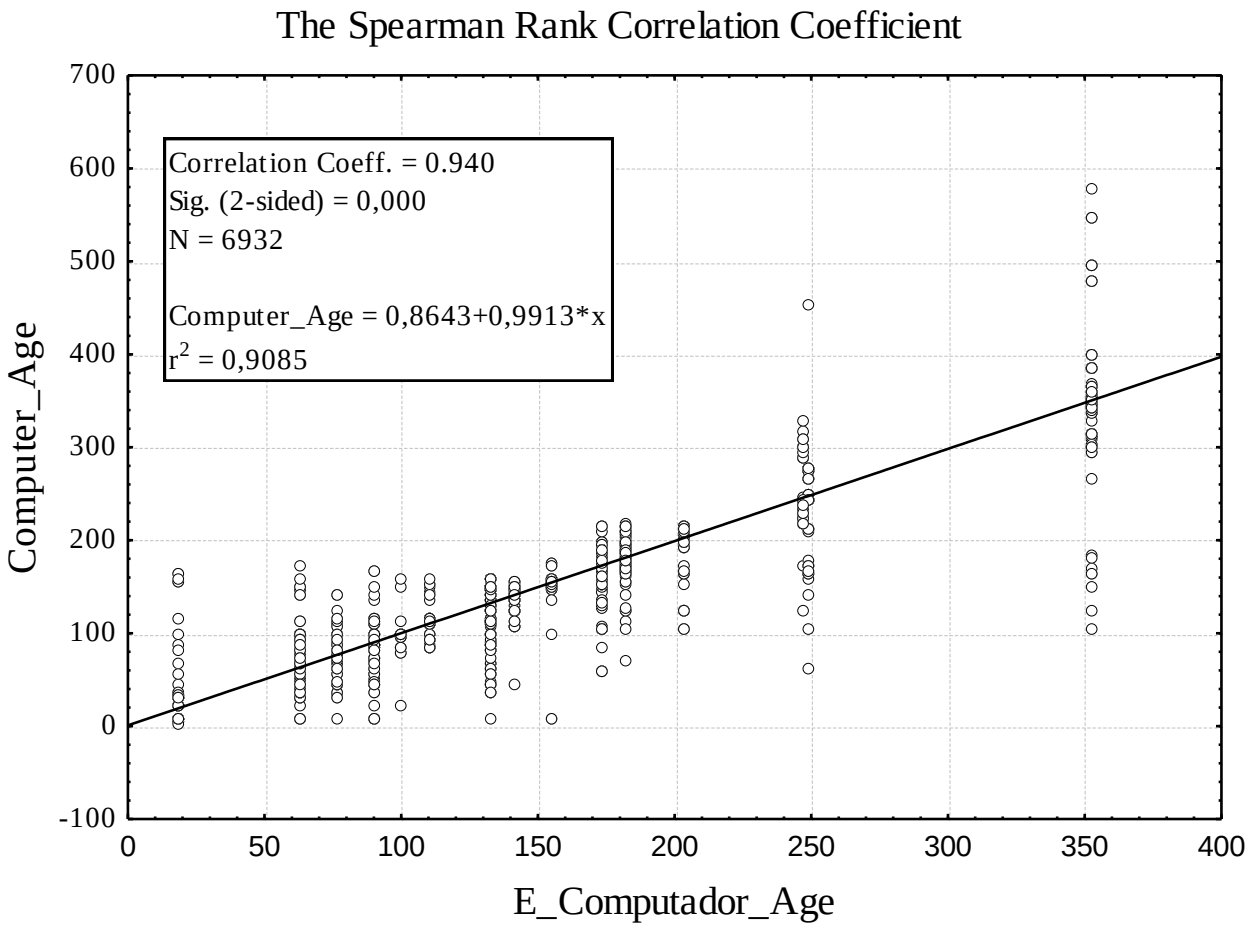


Fig. 2. Scatterplot for *Computer_Age* vs *E_Computer_Age* and Spearman Rank Correlation Coefficient

TABLE VI
COMPUTER 0022 – CMDB PARAMETERS

Variable	Meaning/value	Units
Class	**Missing**	**NA**
Brand	Missing	NA
Computer_Age	Missing	Week
Processor_Tipe	P27	NA
Processor_Clock	2.19	GHz
Processors	2	Count
Memory (RAM)	2.14	GB
Operation_System	SO_7	NA
Service_Pack	SP_3	NA
Hard_Disk	80.02	GB

IV. ANTIVIRUS SCANNING DOSAGE STATISTICS MODEL

Following the imputation process we implement statistical analysis to assess the influence of these variables over malware levels inside an IS. In this study we believe that malware level depends on variables such as: *Processors* (number of processor in the computer), *Computer_Age*, *Class* and *Browse_Time.*

V. CONCLUSION AND FUTURE WORK

Missing values appear frequently in the real world, especially in business-related databases, such as in an IS inside a banking sector, and the need to deal with them is a vexing challenge for all statisticians and data mining modelers. One of the major contributions of CART was to include a fully automated and effective mechanism for handling missing values. CART Models, as were presented here, are more suitable for handling missing values than imputation through random sampling. The reliability of the CART model was evaluated using standard statistical methodology. We find these statistical tools useful to do so.

As a consequence we recommend them.

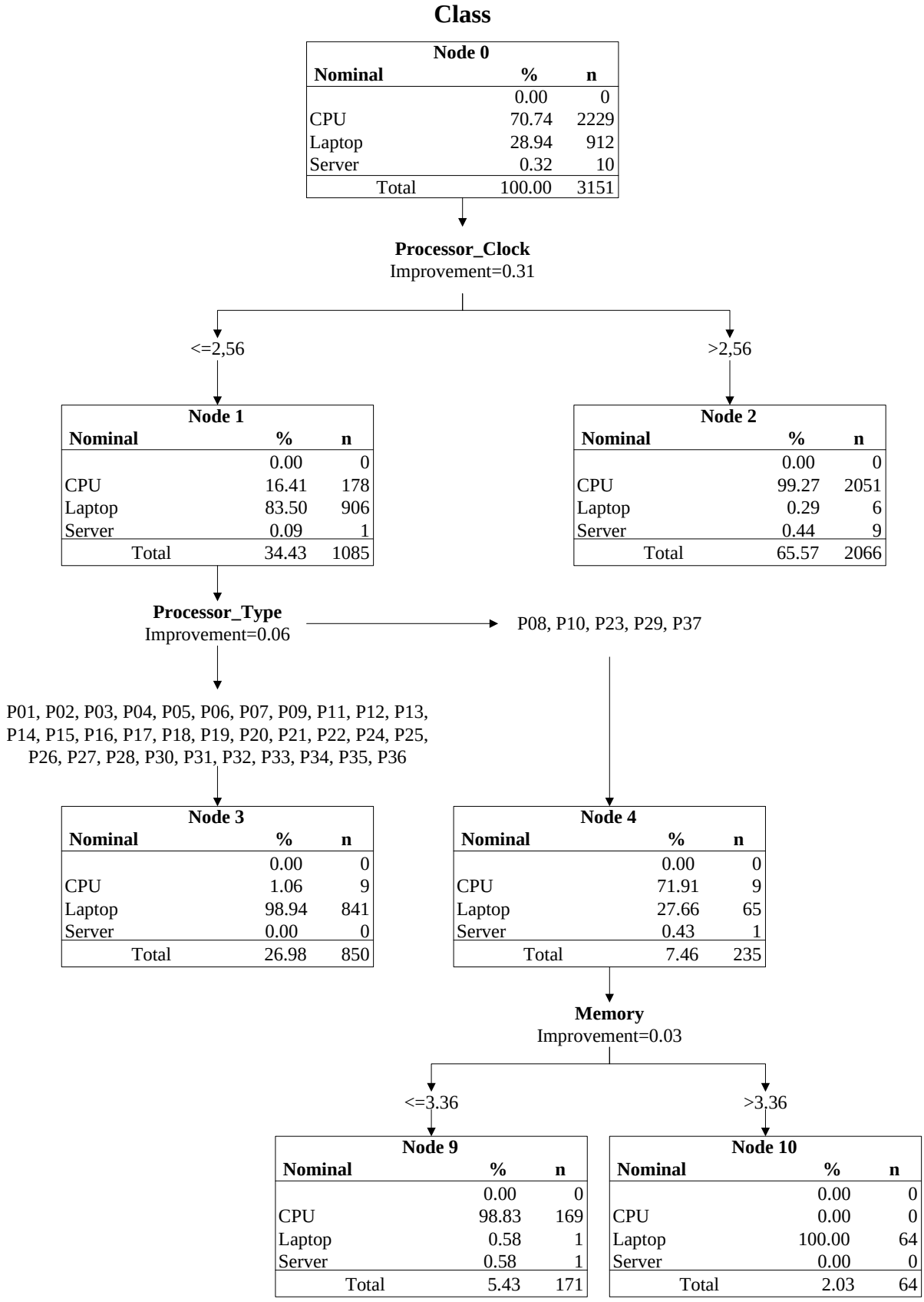


Fig. 3. CART Model for *Class*.

After an adequate imputation process is done, we can use both standard statistical modeling and data mining to explore the association between different variables and the malware levels in an IS. It is important to highlight that we used real data to perform all the analysis. These results may affect the malware scanning policy in a bank.

Currently, data mining efficiently integrate large amounts of data stored in repositories and allows us to discover meaningful new correlations, patterns and trends by using pattern recognition. This is a competitive research advantage for business companies. The statistical and mathematical techniques are essential to data mining for built and check models, it means, statistical tests give more confidence on the results from data mining models. Thus in our case, we evaluated the quality of CART model for handling missing values with different nonparametric statistical tests. The observed results favored this statistical strategy.

Future directions of this work include performing additional statistics analysis such as recurrence analysis and formulation of survival models through Cox-Models. This also will allow identifying significant variables to optimize the malware scanning policy in an IS as well as measure its effect size.

ACKNOWLEDGMENT

The authors wish to thank Juan Carlos Correa from School of Statistics of the Universidad Nacional de Colombia at Medellín for helpful feedback that contributed to improve this manuscript. Also the authors thank the Security Team of the Bank Company for their continuous encouragement and support.

REFERENCES

[1] Weiguo J, "Applying Epidemiology in Computer Virus Prevention: Prospects and Limitations", 2010. Thesis, Computer Science, University of Auckland.

[2] Bailey, N.J.T, "The Mathematical Theory of Infectious Diseases and Its Applications" 1975, New York: Oxford University Press.

[3] Kephart J, and White S, "Directed-Graph Epidemiological Models of Computer Viruses", *IEEE Computer Symposium on Research in Security and Privacy, Proceedings*, pp. 343–359, May 1991.

[4] Kephart J, and White S, "Measuring and Modeling Computer Virus Prevalence, Research in Security and Privacy", 1993, Proceedings, 1993 *IEEE Computer Society Symposium* on, pp. 2–15, May 1993.

[5] Kephart, J, "How Topology Affects Population Dynamics" in *Langton, C.G. (ed.) Artificial Life III*. Reading, MA: Addison-Wesley, 1994.

[6] Pastor-Satorras, R. and Vespignani, A, "Epidemic Dynamics and Endemic States in Complex Networks". Barcelona, Spain: Universitat Politecnica de Catalunya, 2001.

[7] Rishikesh P, "Using Plant Epidemiological Methods To Track Computer Network Worms", 2004. Thesis, Computer Science, Virginia Polytechnic and State University.

[8] Daniel T. Larose, "Discovering Knowledge in Data. An introduction to data mining" 2005. John Wiley & Sons, Inc

[9] Leo Breiman, Jerome Friedman, Richard Olshen, and Charles Stone, "Classification and Regression Trees", 1984. Chapman & Hall/CRC Press.

[10] Vipin Kumar, "The Top Ten Algorithms in Data Mining", 2009. Chapman & Hall/Crc.

[11] Quinlan, R, "Unknown attribute values in induction". *In Proceedings of the Sixth International Workshop on Machine Learning*, 1989 pp. 164–168.

[12] Conover, "Practical Nonparametric Statistics", 1999. John Wiley & Sons, Inc

SPARCL: An Improved Approach for Matching Sinhalese Words and Names in Record Clustering and Linkage

Gayan Prasad Hettiarachchi, Dilhari Attygalle
Department of Statistics, University of Colombo
Colombo-07, Sri Lanka

Abstract- *Quality of data residing in a database gets degraded and leads to misinterpretation due to a multitude of factors. In some cases this results in duplicate records that needs to be merged into a single entity. In doing so, one aspect requirring attention is the way in which string attributes are to be compared. Even though there are different methods in the literature that address the issue of approximate string matching, they all fall short in terms of accuracy when encountered with words from the Sinhalese language written in English. In this paper, it is intended to propose the development of an improved phonetic matching algorithm, which improved the accuracy of approximate string matching remarkably. This algorithm outperforms the phonetic matching algorithms available in the literature when applied on datasets containing Sinhalese names and words written in English. In addition, it demonstrates a computational time comparable with phonetic matching algorithms available in the literature.*

Keywords- Record Linkage, Phonetic Matching, Data Mining, Algorithm Design and Development, Clustering

1. Introduction

Quality of data residing in a database gets degraded and leads to misinterpretation of information due to a multitude of factors. Such factors vary from poor database design (update anomalies due to lack of normalization), lack of standards for recording database fields (person name and address) to typing mistakes (lexicographical errors, character transpositions). Data of such poor quality could result in many damages being caused, for example in a business application, sending wrong products and invoices to the same customer, sending products or bills to wrong addresses, inability to locate customers, etc. In such a case it is important to identify duplicates and merge them into a single entity, i.e. identify whether two entities are approximately the same. In the scientific community this process is known as record linkage [12].

A more formal definition of record linkage can be given as the task of identifying records corresponding to the same entity from one or more data sources. Real world entities of interest include individuals, families, organizations, geographic regions, etc while applications of record linkage are in areas such as marketing, customer relationship management (CRM), law enforcement, fraud detection, epidemiological studies and government administration [13].

Methods used to tackle record linkage problems fall into two broad categories: One commonly used method is deterministic models in which sets of often very complex rules or production systems are used to classify pairs of records as links (i.e. relating to the same entity). The other is the probabilistic model in which statistical or probabilistic methods are used to classify record pairs. In recent years, rapid developments of computational statistics have enabled researchers to move from classical probabilistic methods to newer and advanced approaches using maximum entropy, machine-learning techniques such as Artificial Neural Networks (ANN) and Phonetic matching [13]. Moreover, recent developments in the science of record linkage emphasize on approximate string matching more than any other aspect [3], [15].

The rationale behind focusing attention on approximate string matching is mainly driven by the fact that information about real world entities is most often represented as a collection of string attributes. The enabling technology that breathes life into duplicate identification of string attributes is phonetic matching. In order to perform this matching operation, there are different phonetic matching algorithms available in the literature. They provide a simple and time-tested mechanism for phonetic string matching. Although the simplicity of the design and implementation encourage such an approach in order to develop record linkage applications, limitations of the same are quite significant. The most highlighted drawback of phonetic matching algorithms available in the literature is its limited scope and low accuracy when encountered with words from the Sinhalese language written in English. This is probably due to the fact that phonetic matching algorithms such as Soundex, NYSIIS and Metaphone are developed for English words and it has language components, for example arrangement of vowels and consonants different from Sinhalese language. Therefore, the requirement arises for a modified version of phonetic matching algorithms to suit Sinhalese words and names. The intension of this paper is to present the development of an improved version of Soundex algorithm to suit Sinhalese names and words.

We begin by briefly describing the problem domain and the necessary background on phonetic matching algorithms. Then we describe the proposed algorithm followed by the experimental setup and the results. Finally, we bring into focus a real world application where the improved algorithm can be directly applied in order to increase performance and accuracy.

2. Problem Domain

In many matching situations, it is not possible to compare two strings exactly (character-by-character) because of

typographical errors. Dealing with typographical errors via approximate string comparison has been a major research area in computer science [1]. In record linkage, one needs to have a function that represents approximate agreement, with agreement being represented by 1 and degrees of partial agreement being represented by numbers between 0 and 1[2]. Having such methods is crucial for correct and accurate matching. For instance, in a major census application for measuring undercount, more than 25% of matches would not have been found via exact character-by-character matching [3]. Therefore, a mechanism to perform approximate string matching for datasets with typographical errors is essential in order to achieve high accuracy. The domain of the problem at hand primarily falls under the disciplines of phonetic matching, approximate string comparisons and algorithm development. However, implementing the solution requires in-depth study into several other domains of science and technology such as text retrieval, information retrieval, phonemes, etc.

Phonetic matching is used to evaluate the similarity of pronunciation of pairs of strings, independent of the characters used in their spelling. Queries to sets of strings, in particular databases of names, are often resolved by phonetic matching techniques. For example, when querying a lexicon, only the sound of a string may be known, and in addition, collections of names or words frequently contain spelling, typographical, and homonymic errors making it difficult, if not impossible, to perform one-to-one matching of strings. Thus, the requirement arises for a practical phonetic matching technique. Not only must the algorithm provide reliable judgment of similarity, but must also permit rapid evaluation of queries on a large data set: for example, lexicons of text databases can have vocabularies in excess of one million words [7].

In the literature there are several algorithms for phonetic matching, such as Soundex [6] and the more recent Phonix [4, 5]. These algorithms, which are based on the assumption that the alphabet can be partitioned into sets of sound alike characters are cheap to evaluate but do not perform well when encountered with words from the Sinhalese language. In general, the sound of a word can be described by a sequence of phonemes, which are the basic sounds available for vocalization [7]. A string of phonemes is the pronunciation of the word it represents, or for brevity, it represents sound, as distinct from the word's spelling, or string of letters. The set of phonemes is an international, language-independent standard [8]. A precise phonetic matching algorithm would regard two strings as identical if their sounds were identical, regardless of their actual spelling. It would recognize the similarity of kw and qu and of x and ecks. However, for Sinhalese, and indeed for most languages, phonemes correspond to neither individual letters nor syllables. In general it is not possible therefore, to partition a string or a sequence of letters into substrings that correspond to phonemes; nor is there any possibility of denoting phonemes in terms of sequences of individual letters [14]. In light of the above facts, it is not surprising that phonetic matching algorithms available in the literature do not perform well against Sinhalese words and names. Individual letters of Sinhalese words represented in English do not represent phonemes and many letters have very different sounds in different contexts [14]. Our aim in this research is to discover whether a modified approach to phonetic matching using Soundex might yield better performance, whether phonetic matching together with other algorithms and resources could produce accurate and better results.

3. Phonetic Matching Algorithms

Phonetic matching algorithms focus on the pronunciation of the words instead of the spellings to identify matches. Under phonetic matching, the most profound and time-tested algorithms are Soundex, NYSIIS and Phonix algorithms. A brief description of Soudex will be provided in the following section.

3.1 Soundex Algorithm

Soundex is a phonetic algorithm for indexing names by sound as pronounced in English. The goal is for names with the same pronunciation to be encoded to the same representation so that they can be matched despite minor differences in spelling [6]. Improvements to Soundex are the basis for many modern phonetic algorithms [9]. The Soundex code for a name consists of a letter followed by three numbers: the letter is the first letter of the name, and the numbers encode the remaining consonants. Similar sounding consonants share the same number, for example, B, F, P and V are all encoded as 1. Figure 1 illustrates the Soundex algorithm with respect to the different steps it contains.

- Retain the first letter of the string
- Remove all occurrences of the following letters, unless it is the first letter: a, e, h, i, o, u, w, y
- Assign numbers to the remaining letters (after the first) as follows:
  - b, f, p, v = 1
  - c, g, j, k, q, s, x, z = 2
  - d, t = 3
  - l = 4
  - m, n = 5
  - r = 6
- Remove all pairs of digits which occur beside each other from the string that resulted after the previous step
- Return the first four characters, right-padding with zeroes if there are fewer than four

Figure 1: Soundex Algorithm

4. SPARCL: The Proposed Algorithm

As described earlier in the paper the original Soundex algorithm does not perform well when encountered with words and names from Sinhalese language. This is due to the fact that Soundex, Metaphone and other algorithms were originally designed for words from English language and in addition, Individual letters of Sinhalese words represented in English do not represent phonemes and many letters have very different sounds in different contexts. Therefore, the rationale behind the development of the proposed algorithm is to make

use of the concept of phonemes and consonants from Sinhalese language to increase the accuracy of Soundex for matching Sinhalese names and words represented in English. In doing this, the Soundex algorithm given in Figure 1 had to be modified to represent consonants and phonemes from Sinhalese language. The name SPARCL stands for Sinhalese Phonetic Algorithm for Record Clustering and Linkage. Figure 2 illustrates the modified version of the Soundex algorithm. In addition, the modified algorithm was combined with a string similarity measure that takes into account the primitive number of operations that is required to transform one string to another. This addition, surprisingly, increased the accuracy of the algorithm significantly.

The new addition was facilitated by the Levenstein distance algorithm [9], which compares strings according to spelling alone with no reference to phonetic relationships [10].

One important finding is that distance measures on its own perform better than the original Soundex algorithm when applied on Sinhalese names and words. Despite the accuracy gain, distance measures alone are, however, unable to identify strings with similar sound yet dissimilar spelling such as file and phial or "Nihal" and "Neil". It is the existence of such pairs that motivates the need for a combination of distance measures and a good phonetic matching algorithm.

One way to implement distance algorithms is to measure the closeness in terms of the number of primitive operations necessary to convert the string into an exact match. To be more precise, let P be a pattern string and T a text string over the same alphabet. The Levenstein distance between P and T is the smallest number of changes sufficient to transform a substring of T into P, where the changes may be:

Substitution - two corresponding characters may differ: Rathnapure ⟶ Rathnapura.
Insertion - we may add a character to T that is in P:
Ratnapura ⟶ Rathnapura.
Deletion - we may delete from T a character that is not in P:
Ratthnapura ⟶ Rathnapura.

The algorithm is implemented using a dynamic programming approach that calculates the number of edits D between every possible left-sided substring of each of the two words a and b. $D(a_i, b_j)$, for example, is the edit distance between the first i letters of the word a and the first j letters of the word b. The dynamic programming calculation is recursive, where C_I, C_M, and C_D, are the costs of insertion, substitution, and deletion respectively. In the simplest case, the costs of insertion, deletion, and substitution are all unit costs, and the cost of a match is zero. That is, $C_I(x) = C_D(x) = 1$ for all x and $C_M(x,y) = 0$ if x = y, 1 otherwise.

$$D(a_i, b_j) = \min \begin{pmatrix} D(a_i, b_{j-1}) + C_I(b_i) \\ D(a_{i-1}, b_{j-1}) + C_M(a_i, b_i) \\ D(a_{i-1}, b_j) + C_D(a_i) \end{pmatrix} \quad (1)$$

Results of the experiments carried out on the combined approach will be presented in the next section. It is worthwhile discussing even though it has not been implemented yet, another possible extension that could further improve the accuracy of the proposed algorithm. It is clear that there is no exact algorithm for deriving the likely sound of a string. However, a statistical approach can be adopted to realize this purpose. In the literature there are table books that provide phonemes, phoneme strings and spellings that include the corresponding sounds, but these sources do not provide statistics of the likelihood of correspondence [11]. An alternative approach is to use a software dictionary that provides the spellings and pronunciation of words. The sounds or the pronunciation given in the dictionary provides an alternative approach to phonetic matching. The purpose of phonetic matching can be directly substituted by this method. Adopting this approach, together with distance measures can provide accuracy comparable to the modified approach.

- Retain the first letter of the string
- Remove all occurrences of the following letters, unless it is the first letter: a, e, h, i, o, u, w, y
- Assign numbers to the remaining letters (after the first) as follows:
  - b, f, p, v = 1
  - c, g, j, k, z = 2
  - d -> j if in -dge-, -dgy- or -dgi-
  - q -> k
  - s -> x (sh) if before "h" or in -sio- or -sia-
  - s otherwise
  - s -> x
  - t -> x (sh) if -tia- or -tio-
  - t -> o (th) if before "h"
  - t -> t otherwise
  - l = 4
  - m, n = 5
  - r = 6
  - x -> ks
  - z -> s
- Remove all pairs of digits which occur beside each other from the string that resulted after the previous step
- Return the first four characters, right-padding with zeroes if there are fewer than four

Figure 2: Proposed SPARCL Algorithm

5. Implementation

The implementation of the SPARCL algorithm was carried out in C# programming language under Microsoft Visual Stdio.NET development environment.

The code of the algorithm is organized to optimize the matching process by avoiding unnecessary execution of loops, recursion and redundant comparison of strings. In addition, the code conforms to a comprehensive coding standard enforcing best practices and avoiding pitfalls. The Graphical User Interface (GUI) provides functionality to select any word list stored at a particular location. In addition, functionality is provided to easily apply the SPARCL algorithm on a lexicon and test the accuracy and computational time of the process. The outputs provide the similar pairs of words and names as identified by the algorithm. Furthermore, Soundex algorithm and the Levenstein distance algorithm can also be applied on the specified lexicon in order to carry out a performance

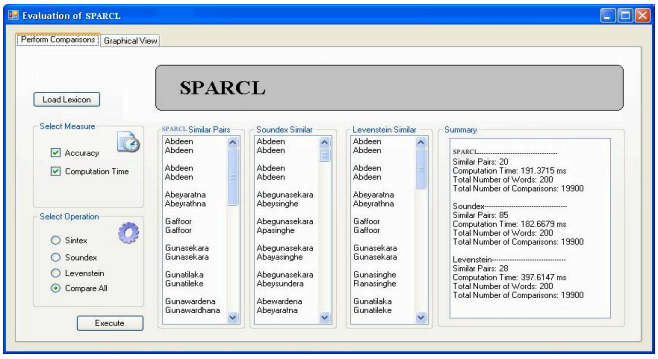


Figure 3: Outputs Obtained by the Algorithms

comparison between the three algorithms. Figure 3 illustrates the outputs obtained by executing the three algorithms on a dataset containing 200 Sinhalese names. Further explanation of the outputs will be provided in section 6.

6. Empirical Evaluation

We now describe our empirical evaluation of the SPARCL algorithms' accuracy and computation time. The datasets that have been used for evaluating SPARCL are described below.

6.1. Datasets

For measuring the accuracy of the SPARCL algorithm we used a dataset containing 200 Sri Lankan surnames. All entries were distinct except for 20 similar surnames with typographical errors, which were deliberately incorporated into the dataset. In addition, a different dataset with 10000 entries were created to compare the computational time of the proposed SPARCL algorithm against the original Soundex algorithm. In addition to comparing the improved algorithm with the original Soundex algorithm, it was compared with the Levenstein distance algorithm.

6.2. Experimental Setup

The accuracy of the SPARCL algorithm was compared with the original Soundex algorithm as well as the Levenstein distance algorithm. This was accomplished using the dataset containing 200 Sri Lankan surnames. Each word in the dataset was compared with the rest of the words only once and any similar pair of words was counted as a match.

Similarly, for testing the computation time of the SPARCL algorithm a dataset of 10000 words was utilized. Each time the SPARCL algorithm was executed on a subset of the dataset. Similarly, the original Soundex algorithm and the Levenstein distance algorithm was executed on the same subset and the computation time or the running time of the algorithms were measured in milliseconds. The obtained results will be discussed in the following sections.

6.3. Accuracy

The SPARCL algorithm produced the same set of similar word pairs that are actually present in the dataset. As described above under section 6.1, we deliberately incorporated 20 pairs of similar words with typographical mistakes. The SPARCL algorithm exactly produced the same 20 pairs of words as matches. Original Soundex algorithm on the other hand, produced 68 incorrect pairs in addition to 17 correct pairs. Similarly, the Levenstein algorithm produced 12 incorrect pairs in addition, to 16 correct pairs. Therefore, SPARCL algorithm demonstrates an accuracy of 100% and an error rate of 0%. (We note that this 100% accuracy was obtained for a dataset with 200 records). Soundex algorithm demonstrates an accuracy of 85%. However, this high accuracy rate is highly compromised by the fact that it produces another additional 68 incorrect matches. Similarly, Levenstein distance algorithm demonstrates an accuracy of 80% with additional 12 incorrect matches. It is clear from the results that Levenstein algorithm alone performs better than the original Soundex algorithm when encountered with Sinhalese words. However, SPARCL, which combines a modified version of Soundex and Levenstein distance algorithms to suit Sinhalese names and words, outperforms the other two methods by quite a margin.

6.4. Computational Time

We measured the computation time for all three algorithms separately using a dataset of 10000 names. The algorithms were run on Windows XP using a 1.66 GHz Intel Centrino Duo machine with 512 MB of RAM. Different subsets of the dataset were created ranging from 200 entries to 10000 entries. For all subsets, computation times were within an order of magnitude of each other for all three algorithms. Original Soundex algorithm is faster than both the proposed algorithm and the Levenstein distance algorithm. However, the proposed approach and the original Soundex produced comparable results with respect to computational time. Even though, Soundex demonstrates a small performance edge over the proposed algorithm, it is highly compromised by the huge reduction in accuracy when encountered with words from Sinhalese language. The computational times for each of the subsets are given in Table 1. Figure 4 illustrates a graphical representation of the computational times of the three algorithms in the same chart.

One area where duplication is quite evident and record linkage proves to be quite useful is with regard to newspaper articles. We consider such articles related to human right violations. Articles related to the same incident are published in various newspapers on different days in various ways. Also the same incident is reported and discussed in the same newspaper continuously for several days. This leads to the problem of duplication which can in turn lead to other large scale problems at the national and international level. This type of duplication can reveal an incorrect image about the country's situation to local as well as international communities.

Such duplication can lead to an over estimation of the violations taken place in a country and could be even more severe when there are ethnic issues or a civil war going on in a country like Sri Lanka. Record linkage can help alleviate these problems to a great extent. A dataset of human right violations would generally consist of attributes such as victim name, location, incident type, perpetrator, etc. These attributes are often represented as character strings. In order to perform

Table 1: Comparison of Computational Times

Number of Words	Computational time in milliseconds		
	SPARCL algorithm	Original Soundex algorithm	Levenstein distance algorithm
200	105.8	94.6	127.8
500	613.3	587.1	807.1
1000	2506.8	2371.5	3241
2000	9634.5	9570	13255.5
3000	21668.5	21560	29942.8
4000	39001.9	38866.5	54619.8
5000	61864.6	61384.2	89290.2
6000	89166.4	87390.9	129449.9
7000	121870	121321.9	172661.9
8000	160682	159197.5	224700
9000	205752.8	207589.3	292965.4
10000	253989.5	255694.7	362773.3

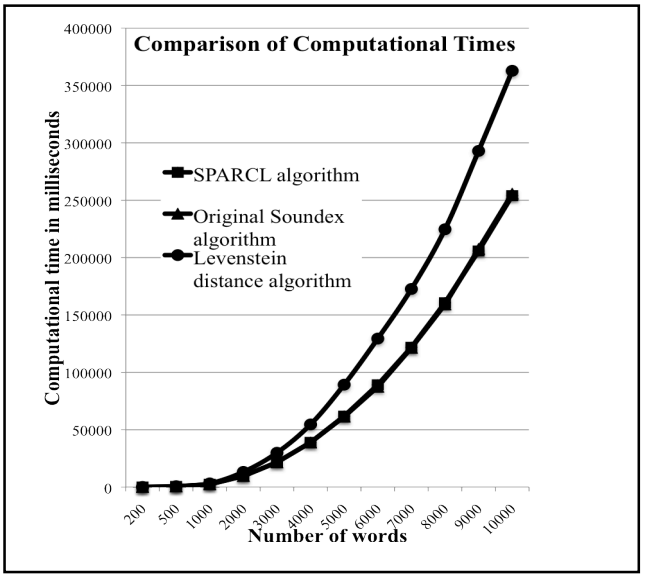


Figure 4: Comparison of Computational Times: Both SPARCL and Soundex algorithms show comparable results with respect to computational time. Therefore, lines corresponding to these two algorithms are almost overlapped in Figure 4.

record linkage on these attributes we can make use of phonetic matching. Since these attributes contain names and words from Sinhalese language, the original Soundex algorithm as mentioned earlier is unable to provide the required level of accuracy.

However, according to results given in section 6, the proposed SPARCL algorithm can provide the required foundation to achieve high level of accuracy. In addition, the use of SPARCL algorithm for this purpose will not result in a huge performance loss due to the fact that it demonstrates a comparable computation time with the original Soundex algorithm.

Apart from the aforementioned purpose, the SPARCL algorithm can also be used in the development of a generic framework for record classification and linkage. The SPARCL algorithm will lie in the center of the framework providing the backbone for duplicate identification of string attributes. In addition, the framework can provide additional functionality such as clustering, blocking, weighting attributes, selection of blocking variables, prediction of missing attribute values, etc required by a general record linkage program. The development of the framework is under working progress.

7. Conclusion

This paper proposes a modified phonetic matching algorithm as an alternative to both Soundex algorithm and the Levenstein distance algorithm for comparing names and words from Sinhalese language written in English. Experiments show that SPARCL produce better results in terms of accuracy than the other two approaches. In terms of performance, both SPARCL algorithm and Soundex algorithm show similar computational times on a number of datasets. However, Levenstein distance algorithm shows very low performance relative to the other two approaches.

Directions for future research include refining the algorithm to make use of the pronunciations provided in an online or offline dictionary to perform phonetic matching. In addition, the proposed SPARCL algorithm can be applied to a myriad of real world problems including the development of a framework for record classification and linkage.

References

[1] Hall, P. A. V., and Dowling, G. R. (1980), "Approximate String Comparison," Computing Surveys, 12, 381-402.

[2] Winkler, W. E. (1995), "Matching and Record Linkage," in B. G. Cox et al. (ed.) Business Survey Methods, New York: J. Wiley, 355-384.

Verykios (2007), "Duplicate Record Detection: A Survey" , IEEE Transactions on Knowledge and Data Engineering 19 (1): pp. 1–16

[4] T.N. Gadd. (1988), `Fisching fore werds': Phonetic retrieval of written text in information systems. Program: automated library and information systems, 22(3):222-237.

[5] T.N. Gadd. (1990), PHONIX: The algorithm. Program: automated library and information systems, 24(4):363

[6] P.A.V. Hall, G.R. Dowling (1980), Approximate string matching. Computing Surveys, 12(4):381{402, 1980.

[7] A.C. Gimson and A. Cruttenden (1994), Gimson's Pronounciation of English. Edward Arnold, London, _fth edition.

[8] (2007), The principles of the International Phonetic Association. Phonetics Department, University College, London, UK.

[9] J. Zobel and P. Dart. (1995), Finding approximate matches in large lexicons. Software Practice and Experience, 25(3):331.

[10] K. Kukich. (1992), Techniques for automatically correcting words in text. Computing Surveys, 24(4):377.

[11] Nielsen, Sandro (2008), "The effect of lexicographical information costs on dictionary making and use", in Lexikos (AFRILEX-reeks/series 18), pp.170–189

[12] G. Salton and M.J. McGill. Introduction to Modern Information Retrieval. McGraw-Hill, New York, 1983.

[13] Alvey, W. and Jamerson, B. (eds.) (1997), Record Linkage Techniques -- 1997 (Proceedings of An International Record Linkage Workshop and Exposition, March 20-21,1997, in Arlington VA), Washington, DC: Federal Committee on Statistical Methodology.

[14] Dulip Herath, Kumudu Gamage, Anuradha Malalasekara, Research Report on Sinhala Lexicon. [Online]. Available: http://www.panl10n.net/english/final%20reports/pdf%20files/Sri%20Lanka/SRI01.pdf
[Accessed: Apr, 29, 2008]

[15] Rohan Baxtor, Peter Christen, Tim Churches (2003), A Comparison of Fast Blocking Methods for Record Linkage, Workshop on Data Cleaning, Record Linkage and Object Consolidation, 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington DC

The Importance of Tuning Financial Technical Indicators to Predict Stock Movements

Maysa Ammouri[1] and Sameh Al-Shihabi[2]
[1]Industrial Engineering Department, German-Jordan University, Amman, Jordan
[2] Industrial Engineering Department, University of Jordan, Amman, Jordan

***Abstract*— *Different models that rely on financial technical indicators to predict stock movements have been suggested by researchers. These models, however, ignore the importance of selecting the proper time windows to be used by these indicators. In this work, we choose the time windows used in these indicators by maximizing the Fisher Discriminant Ratio (FDR). Fifteen commonly used indicators are then employed as inputs to a Naïve Bays (NB) classifier that is trained to predict the stock movement directions of ten companies for the period from 1/1/2000 to 1/2/2012. The accuracy of the NB classifier has improved by around 20% using our suggested time windows compared to the default values that are used in practice.*

Keywords: Technical Indicators, Fisher Ratio, Stocks

1. INTRODUCTION

THE use of technical indicators to predict stock movements has been addressed by many researchers and applied by practitioners. Researchers use these indicators along with other data mining tools to predict the direction of stocks' movements or their indices [1]. Based on information offered by these indicators, practitioners develop trading rules that tell them whether to hold, sell, or buy the stocks [2]-[4].

Using these indictors to predict the stock movement for next day would mean relying on information obtained from previous trading days. Some of these indicators would depend on one day earlier; others, however, would depend on the previous $n>1$ days. Recommendations are usually given regarding the number of days, time windows, to consider for calculating these indicators. For example, the financial tool box in Matlab, http://www.mathworks.com, recommends using 10 periods to calculate Chaiken Volatility and 12 periods to estimate momentum [5].

The accuracy of any classifier that uses these indicators would largely depend on the quality of information provided by them. For this purpose, fine-tuning these indicators is important, which is the theme of this paper.

To tune these indicators, we try to maximize the Fisher Discriminant Ratio [6] (FDR), where the decision variable considered is the number of time periods needed to calculate the indicators. To check the validity of our hypothesis, we built a classification model to predict the stock price movements. The rule is a Naïve Bays [7] (NB) algorithm. The accuracy of the NB algorithm that uses tuned time windows is compared to a similar classifier that uses default time windows, as recommended by practitioners. The NB classifier attempts to predict the movement direction for 10 stocks for the period from 1/1/2000 to 30/5/2012. The improvement obtained by tuning these indicators was more than 30%.

The importance of this work stems from the fact that tuning technical indicators is ignored in the research literature; however, it has a significant impact on any predication or classification algorithm. For instance, in [4], the time windows used to calculate the technical indicators are not mentioned, which makes the reader assume that the default values are used. On the other hand, the time windows to use are found via a trial-and-error methodology in [8]. Additionally, technical indicators to predict stock indices and intra-day prices are addressed more commonly in the literature than the closing prices of individual stocks. We show here how a simple model receiving its input from well-tuned indicators can have high predictability over a number of stocks.

2. Technical Indicators

Numerous technical indicators have been suggested in past decades; however, not all of them gained the same reputation as being good measuring sticks for stocks' behaviors. In this work, we only select indicators that are parametric: controlled by a time window parameter. We used the indicators coded in Matlab, http://www.mathworks.com, in addition to a number of other indicators cited in the research papers [1, 4, 8]. Table 1 lists these indicators, where Column One shows their abbreviated names, followed by their names and their respective formulas.

TABLE 1
Financial Technical Indicators

Symbol	Name	Formula

SMA	Simple Moving Average	$\sum_{i=0}^{i=n-1} \frac{C_{t-i}}{n}$
EMA	Exponential Moving Average	$(C_t - EMA_{t-1}(C)).\frac{2}{1+n} + EMA_{t-1}(C)$
CV	Chaiken Volatility	$\frac{\left\{ EMA_t(H-L) \atop -EMA_{t-n}(H-L) \right\}}{EMA_n(H-L)}$
HH	Highest High	$\max\{H_{t-1}, H_{t-2}, \ldots, H_{t-n}\}$
LL	Lowest Low	$\min\{L_{t-1}, L_{t-2,\ldots,} L_{t-n}\}$
WILL%R	William9s %R	$\frac{H_n - C_t}{H_n - L_n} * 100$
MFI	Money Flow Index	$TP_t = \frac{H_t + L_t + C_t}{3}$ $MF_t = TP * Volume$ $MFR_t = \frac{+ve \text{ MF(n)}}{-ve \text{ MF(n)}}$ $100 - \frac{100}{1 + MFR_t}$
VOLROC	Volume Rate of Change	$\frac{V_t - V_{t-n}}{V_{t-n}}$
MOM	Momentum	$C_t - C_{t-n}$
ROC	Rate of Change	$\frac{C_t}{C_{t-n}} * 100$
RSI	Relative Strength Index	$RS = \frac{\frac{\sum_{i=1}^{n} Up_{t-i}}{n}}{\frac{\sum_{i=1}^{n} Dw_{t-i}}{n}} * 100$ $100 - \frac{100}{1.0 + RS}$
%K	Fast Stochastic %K	$\frac{C_t - LL_{t-n}}{HH_{t-n} - LL_{t-n}} * 100$
%D	Fast Stochastic %D	$\sum_{i=0}^{n-1} \frac{\%K_{t-i}}{n}$
S%D	Slow Stochastic %D	$\sum_{i=0}^{n-1} \frac{\%D_{t-i}}{n}$
MACD	Moving Average Convergence Divergence	$EMA_{n3}\left\{ EMA_{n1}(C_t) \atop -EMA_{n2}(C_t) \right\}$

where C_t, L_t and H_t represent the closing price, lowest price and highest price in day t, respectively. On the other hand, Up_t represents the upward closing price change in day t, while Dw_t is the downward closing price change. The $+ve\ MF(n)$ is the summation of MF the past n days when $TP_t - TP_{t-1} \geq 0$, while $-ve\ MF(n)$ is the opposite.

As noticed from the formulas above, parameter n controls the value of all the financial indicators; except for MACD since three parameters are needed.

3. Fisher discriminant ratio

The Fisher Discriminant Ratio [6] assumes that data points belong to two different classes. It tries to find a plane such that projecting the data points on this plane maximizes the accuracy of classification. This plane maximizes the distance between the centers of the two projected clusters, intra-distance, while minimizing the distances between the points belonging to the same class, inter-distance.

This ratio is given by

$$J(w) = \frac{(\tilde{\mu}_1 - \tilde{\mu}_2)^2}{\tilde{S}_1 + \tilde{S}_2} \quad (1)$$

where:

$J(w)$: is the Fisher Discriminant Ratio.

w: is the projection plane.

$\tilde{\mu}_c$: is the mean value of the projected points belonging to cluster c=1, 2.

$\tilde{S}_c$: is the variance of projected points belonging to class c=1, 2.

The plane w is found by maximizing the above value, which is given by

$$w = \frac{(\mu_1 - \mu_2)}{S_1 + S_2} \quad (2)$$

where the symbols in the equation stand for the original data, not their projections.

For each financial indicator FI, we try to maximize the value of $J(w)$ corresponding to the indicator FI, denoted by $J_i(w)$. In our work, class 1 or 2 refers to the instance classification based on the stocks' direction the next day. Using a full enumeration approach, we maximize the value of $J_i(w)$ by controlling the time window(s) used by indicator i.

The optimization equation becomes:

$$\max_{n} J_{FI(n)} \quad (3)$$

such that

$$LL \leq n \leq UL \quad (4)$$

.

4. Naïve Bayes Classifier

The stock movement prediction problem can be considered as a binary classification problem. This is due to the fact that when the stock's price remains the same or decreases, then it can have a value of 0, while it will have a

value of 1 if the price increases. A Naïve Bays (NB) classifier is used in this work due to its simplicity.

Assuming that we have two classes, U and D, and one feature F, the NB classifier estimates

$$\Pr[Class = U / \text{Feature} = f] = \frac{\Pr[Feature = f / Class = U]\Pr[Class = U]}{\Pr[Feature = f]} \quad (5)$$

For more than one attribute, this equation becomes

$$\Pr\{U / f_1, f_2, .., f_n\} = \frac{\Pr\{f_1 f_2, ..., f_n / U\} * \Pr\{U\}}{\Pr\{f_1, f_2, ..., f_n\}} \quad (6)$$

As the NB classifier assumes independence among the features, this last equation can be calculated using

$$\Pr\{U / f_1, f_2, .., f_n\} = \frac{\Pr\{f_1 / U\}\Pr\{f_2 / U\}...\Pr\{f_n / U\}\Pr\{U\}}{\Pr\{f_1\}\Pr\{f_2\}...\Pr\{f_n\}} \quad (7)$$

The conditional probabilities $\Pr\{f_i / U\}$ can be calculated for each feature $i = 1,2,..n$. The instance is classified by solving the following optimization model:

$$\max_{Class \in \{U,D\}} \Pr\{Class / f_1, f_2, ...f_n\} \quad (8)$$

As the inputs from the technical indicators are continuous variables, calculating the conditional probabilities can be done by either assuming that the features follow a normal distribution function or after binning the features and dealing with them as a discrete probability distribution function. The former method works as follows: after classifying the instances to the corresponding classes, the mean and variance of the feature, for each class, is calculated. The probability that feature f_i has value Y is calculated.

Given its name, in the binning method, the features are binned after classification, and the conditional probabilities of having a certain class in any bin are estimated. In this work, the binning scheme is adopted to calculate the conditional probabilities.

5. Experimental study

The first step conducted in the experimental part of this work is tuning the technical indicators, as explained earlier. The study shows that the different stocks would require different time windows for the same indicator. These indicators are used with the Naïve Bays classifier to predict stock movements. The indicators would use the default values in one experiment and the tuned values in the other. The difference in performance is then summarized.

The stocks chosen for this study were taken from five different industries, namely, basic materials, conglomerates, consumer goods, financial, and industrial goods. Table 2 shows these stocks, where the second column shows the company name, followed by its industry, and the first column represents the symbol used in the stock market, or the ticker name.

TABLE 2
STUDIED STOCKS

Symbol	Name	Industry
DOW	Dow Chemicals	Basic Materials
RTK	Rentech Inc.	Basic Materials
DHR	Danaher Corp.	Conglomerates
MMM	3M	Conglomerates
PEP	Pepsi Co.	Consumer Goods
PHG	Philips Electronics	Consumer Goods
C	Citigroup	Financial
JPM	JP Morgan	Financial
GE	General Electronics	Industrial Goods
CAT	Caterpillar Inc.	Industrial Goods

5.1. Parameters Tuning

In this experiment, we maximize the value of the FDR using full enumeration. For each indicator, time windows ranging from 2 days to 30 days were tested. Table 3 summarizes the results obtained. It is interesting to note that for some indicators, the optimum values covered the entire range. The SMA, EMA, CV, HH, LL, MFI, and VOLROC indicators show that the time windows used need to be greater than the ones known for investors. The rest of the indicators show that it needs to be as small as possible. For the different companies, different optimum time windows were calculated for the same technical indicator.

TABLE 3
Comparison Between Default and Tuned Time Windows

Symbol	Default	Average	Range
SMA	10	17.1	2-30
EMA	10	16.4	2-30
CV	10	12.9	10-30
HH	14	15.4	6-30
LL	14	24.1	2-30
WILL%R	14	3.8	2-9
MFI	14	8.5	4-25
VOLROC	12	5	0-16
MOM	12	2	2
ROC	12	2	2
RSI	14	7.3	2-29
%K	10	3.7	2-12
%D	3	2	2
S%D	3	2	2
MACD	(12,26,9)	(2,3,2.5)	(2,3,2-3)

5.2 Classification Experiment

Using WEKA software, http://www.cs.waikato.ac.nz/ml/weka/ , an NB classifier is built using the twelve above mentioned indicators as inputs. As stated earlier, we binned the values obtained by each indicator in 10 bins. The experiment conducted divided the data into training data and testing data. One third of the data is used to train the NB, while the other two thirds are used to test the model and to report the results.
In the first experiment, the default time windows were used. This experiment was repeated with the time windows selected in the first experiment.
The following table shows the difference in classification accuracies between the two experiments.

TABLE 4
Naïve Bays Classification Accuracy Using the Default Time Windows and the Tuned Ones

Symbol	Default	Tuned	Difference
DOW	61.42	85.31	23.88
RTK	63.81	96.70	32.89
DHR	65.05	84.26	22.21
MMM	60.01	81.16	21.12
PEP	60.36	85.73	25.37
PHG	61.69	86.28	24.59
C	60.84	89.06	28.22
JPM	59.84	83.40	23.56
GE	61.04	81.05	20.01
CAT	59.68	83.26	23.58

As seen from Table 4, the accuracy of the Naïve classifier has dramatically improved after tuning the financial indicators.

6. Conclusion and Future Research

This work shows the impact of tuning the technical indicators on the accuracy of a simple classifier algorithm. Before using complicated and hybrid classifiers, researchers should tune the indicators when using them. More investigation is needed to support the hypothesis presented in this work. Currently, the impact of tuning these indicators on the accuracy of predicting stock indices is under investigation. The impact of the tuning these indicators on other simple classifiers is being investigated as well.

7. References

[1] G. S. Atsalakis and K. P. Valavanis, "Surveying Stock Market Forecasting Technicques-Part II: Soft Computing Methods," Expert Systems with Applications, vol. 36, no. 3, pp. 5932–5941, April. 2009.

[2] W. Leigh, N. Modani, R. Purvis, and T. Roberts, "Stock Market Trading Rule Discovery Using Technical Charting Heuristics," Expert Systems with Applications, vol. 23, no. 2, pp. 155–159, August 2002.

[3] T. Chavarnakul and D. Enke, "A Hybrid Stock Trading System for Intelligent Analysis-Based Equivolume Charting," Neurocomputing, vol. 72, no. 16-18, pp. 3517–3528, Oct. 2009.

[4] Y. Kara, M. A. Boyacioglu, and Ö. K. Baykan, "Predicting Direction of Stock Price Index Movement Using Artificial Neural Networks and Support Vector Machines: The Sample of the Istanbul Stock Exchange," Expert Systems with Applications, vol. 38, no. 5, pp 5311-5319, May. 2009.

[5] R. Bauer and J. Dahlquist, Technical Market Indicators, Analysis and Performance. New York: Wiley, 1999.

[6] S. Wang, D. Li, X. Song, Y. Wei, and H. li, "A Feature Selection Method based on Improved Fisher's Discriminant Ratio for Text Sentiment Classification," Expert Systems with Applications, vol. 38, no. 7, pp. 8696–8702, August 2002.

[7] I. Witten, E. Frank, and M. Hall, Data Mining: Practical Machine Learning Tools and Techniques. The Morgan Kaufmann Series in Data Management Systems, 2011.

[8] M. Tanaka-Yamawaki and S. Tokoura, "Adaptive Use of Technical Indicators for the Prediction of Intra-day Stock Prices," Physica A, vol. 383, no. 1, pp. 125–133, August 2002.

SESSION

SEGMENTATION, CLUSTERING, ASSOCIATION

Chair(s)

Dr. Robert Stahlbock
Dr. Gary M. Weiss

Cartogram Data Projection for Self-Organizing Maps

David H. Brown and Lutz Hamel
Dept. of Computer Science and Statistics
University of Rhode Island
USA

Abstract— *Self-Organizing Maps (SOMs) are often visualized by applying Ultsch's Unified Distance Matrix (U-Matrix) and labeling the cells of the 2-D grid with training data observations. This does not provide two key pieces of information when considering real world data: (a) While the U-Matrix indicates the location of possible clusters on the map, it typically does not accurately convey the size of the underlying data population. (b) When mapping data onto the the SOM, multiple observations often are mapped onto a single cell of the grid. We address these shortcomings with two complementary visualizations. First, we increase or decrease the 2-D size of each cell according to the number of data elements it contains. Second, we determine the within-cell location of each mapped training observation according to its similarity in n-dimensional feature space to each of the immediate neighbor nodes that surround it on the 2-D SOM grid. When multiple observations are mapped to a single cell then the plot locations will convey a sense of the data distribution within that cell. These techniques lend themselves to additional applications and uses which we demonstrate.*

Keywords- Self organizing feature maps; Data visualization; Data mining; cartogram

1. Introduction

Kohonen's self-organizing maps (SOM) [1] employ an artificial neural network to reduce the dimensionality of an $\Re^n$ dataset while preserving the topology of its data relationships. The SOM network is typically constructed and visualized as a regular two-dimensional grid of cells, each representing a single node. Thus, each node has both a feature-space $\Re^n$ value and a 2-D (x,y) position visualized as a cell in the SOM grid. (When we speak of the neighbors of a node, we mean those nodes whose 2-D grid positions are adjacent to that of the indicated node.)

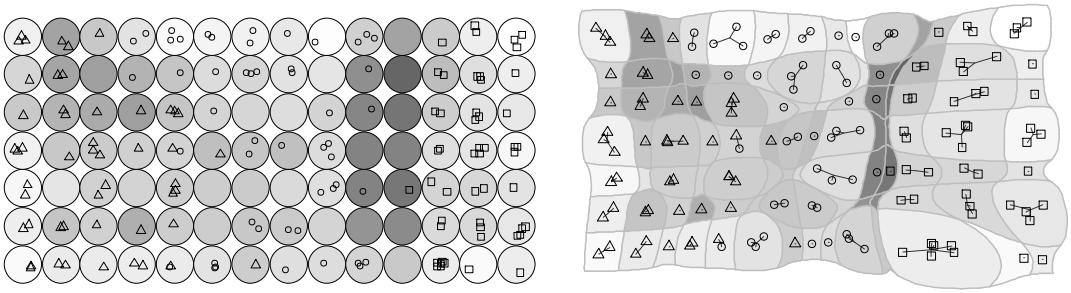
During training, adjustments to each node's *n*-dimensional values are also partially applied to nodes found within a time step sensitive radius of its 2-D grid position. Thus, changes in feature-space values are smoothed, forming clusters of similar values within the local neighborhoods on the 2-D grid.

Clustering is often indicated by shading each cell to indicate the average distance in feature-space of the node to its 2-D grid neighbors; this is the Unified Distance Matrix (U-Matrix) [2]. To map training data to this grid, the node nearest in feature-space to a training observation is identified. This is the "best-match" node for that observation and the observation is plotted in the grid cell of that node [3]. This is done for all training instances. Often, multiple observations map to the same cell in the grid.

This standard visualization of the SOM is a powerful tool for gaining understanding of the overall structure of a dataset, but it can obscure important information about individual data. It does not reliably show the size of the underlying data population within the clusters. The straightforward labeling of cells with their data does not provide any insight into the feature-space distribution of the data within that cell.

To remedy the cluster size representation problem, we expand and contract the 2-D SOM grid cells in proportion to the number of data points plotted in each. This shows clusters in proportion to their population and it also opens up space within the more populous cells for plotting the data more informatively. The resulting plot is called a cartogram and is a technique borrowed from geography [4]. To visualize the data distribution within each cell, we show the feature-space separation between each observation and its corresponding node on the grid. Data points that are most similar to the node (in feature space) appear at or near the center of the 2-D grid cell. Data points that are less similar to the node are moved toward the grid neighbors, which they are most similar to in feature space. The spread of the data around the center of the cell also indicates the quantization error. The quantization error is a measure of "goodness of fit" and is defined as the feature space distance between a training data point and its best matching node on the map [1].

A comparison of the standard visualization and our enhancements is shown in Figure 1. In Figure 1(a) we show a standard U-Matrix visualization of the familiar iris data set [5][6] and in Figure 1(b) we show our enhanced visualization of the same SOM using the cartogram and the visualization of the quantization error. It is easy to see that the maps have three main clusters. In the map in Figure 1(b) the size of the cells have been resized in the proportion of data points mapped to the cells. The data positions within the cells of the map in Figure 1(a) are random jitter and therefore contain no information. In our visualization, Figure 1(b) the within cell positions are meaningful and are computed from the data.



(a) (b)

Figure 1. (a) The typical SOM visualization combining U-Matrix shading with randomly assigned data positions within the cell. (b) Our enhancements. The SOM has been constructed from the Fisher/Anderson "iris" data set; darker shading indicates cluster boundaries (greater feature-space distance between cells). Plot symbols correspond to iris species (class). The linearly separable *setosa* species is on the right (square symbols).

2. Cartogram

A cartogram is a geographical map that has been distorted so that the area occupied by each region of the map corresponds to the value of some parameter related to that region. For example, countries of the world might be shown scaled in proportion to their population, per-capita income, or any other metric of interest. In our cartogram visualization, we have adopted the diffusion method of Gastner and Newman [7].

The goal of a cartogram is to reshape the features of the map so that the average density of the metric of interest – e.g., population – is uniform throughout the map. The diffusion cartogram achieves this by calculating the density gradient at each vertex in the map and moving the vertices along the gradient toward the less-dense area. Areas where the metric is greater than the average are expanded (and so made less dense) and areas where the metric is less than average are reduced in size (increasing their density). This process repeats until the reshaped map stabilizes: all areas are at the average density and so the gradients are zero. (A more complete, two-page description of the algorithm in pseudo-code is available online as a "Supporting Text" to the original paper [8].)

To construct a cartogram for our visualization, we start with the hexagons or rectangles that outline each of the 2-D cells of the SOM map. The initial density within these polygons is calculated as a function of the number of data points mapped to that cell.

Using the 'sp' (Spatial Polygons) package [9][10], we transfer our original map of polygons – hexagons or rectangles – and their population density values to a fine rectangular mesh that can be processed efficiently by the 'Rcartogram' package [11] – an interface to Newman's implementation of the diffusion cartogram algorithm in C [12]. The "cart" object returned is used to translate grid polygon points to their new positions on the cartogram visualization.

3. Data Mapping Within the Cell

To position the each training observation within the cartogram-expanded cell, we begin (as do other visualizations) by selecting its best-match node in feature-space; the observation will appear within that node's cell. Then:

- a feature-space vector from that best-match node to each of its neighbors is calculated,
- the relative length of the orthogonal projection of the training observation along each neighbor vector is calculated in feature-space, and
- a 2-D offset vector is calculated and added to the best-match node's position in the 2-D grid.

The resulting location meaningfully and consistently places the observation on the map.

3.1. Selecting the Best-Match node

The data point to be plotted (x) is compared to each node's feature-space value (m_i) using some metric such as the least Euclidian distance [13],

$$\|x - m_b\| = \min_i \{\|x - m_i\|\}, \text{ or}$$
$$b = \arg\min_i \{\|x - m_i\|\} \quad (1)$$

Node b is the node nearest to the data point in feature space: the best-match node.

3.2. Finding Vectors to Neighbors

The feature-space vectors to each of the j neighbors (m_j') are calculated simply by subtracting the $\Re^n$ feature-space value of the best-match node, m_b, from that of each neighbor m_j (a linear translation,),

$$m_j' = m_j - m_b \quad (2)$$

Likewise, the data point's translated vector is,

$$x' = x - m_b \quad (3)$$

Applying the same translation to the best-match node itself confirms its role as the origin for the calculations that follow (its value is 0 in every coordinate axis),

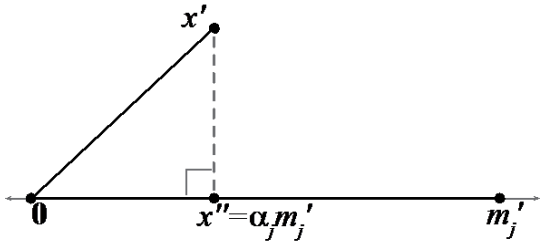


Figure 2. Orthogonal projection (*x*") of a training instance vector (*x*') onto a neighboring node vector (m_j'). These calculation is generalizable to the $\Re^n$ feature space of the SOM [8].

$$\mathbf{0} = m_b - m_b \tag{4}$$

3.3. Orthogonal Projection

In order to determine how far a data point should shift from its best-match node toward each neighbor node, we consider its orthogonal projection onto each neighbor vector in the *n*-dimensional feature space of the SOM. The translated data point vector (*x'*) can be thought of as the sum of two component vectors: one (*x''*) directly along the vector to the neighbor node (m_j') and the other at right angles to the first. The vector *x''* is the *orthogonal projection* of the data point vector onto the neighbor node vector. As shown in Figure 2, the orthogonal projection is equal to the product of some scalar value α_j and the translated neighbor vector.

This α_j value (proportional projection toward the neighbor) is found using the dot product (inner product) of vectors [14],

$$\alpha_j = \frac{x' \cdot m_j'}{m_j' \cdot m_j'} \tag{5}$$

If the data point is on the "other side" of the origin (headed away from a neighbor), the value of α_j will be negative. Neighbors with positive α values will "pull" the data point in their direction on the grid while neighbors with negative α "push" the data point away.

3.4. Calculate and scale the 2-D Offset

Again taking the center of best-match node *b* as the origin (this time in 2-D grid space: g_b), the translated grid coordinates of each neighbor g_j are multiplied by the proportional length α_j and added together to form a raw 2-D offset vector *r*,

$$r = \sum_{j \in \{\text{neighbors}\}} \alpha_j \left(g_j - g_b \right) \tag{6}$$

Typically, several neighbors contribute to this raw offset, exaggerating the data point's distance from its best-match node. For example, if the neighbor to the left has a positive α, pulling the data point to the left, the neighbor to the right might very well have a negative α and push the data point even further to the left. Diagonal neighbors can push or pull along both axes.

If the raw offset is used, the data point will frequently appear outside the area of its best-match node's cell; this incorrectly suggests that some other node is nearest. We have found that the simplest satisfactory scaling function is to divide the raw offset by the number of neighbors surrounding the best-match node,

$$s = r \div \|\{\text{neighbors}\}\| \tag{7}$$

There is an aesthetic and practical tension between ensuring that data points are displayed within the area of their best-match nodes while not limiting offsets to a range too small to be perceptible. Alternate approaches to scaling are possible and continue to be explored but this simple scaling scheme described here seems to work appropriately.

Finally, the scaled offset *s* is added to the 2-D coordinates of the best-match node (g_b), giving the plotted grid position of the datum, g_d,

$$g_d = g_b + s \tag{8}$$

3.5. Visual representation

An appropriate symbol or label is drawn at the position g_d calculated in Equation 8. We also add a thin line connecting each symbol back to the center of its cell, g_b. This visually reinforces the interpretation of the plotted position as a vector with respect to the best-match node.

On occasion, the cartogram reshaping of the map can produce cell outlines where the true center is not immediately obvious. An example may be found in the second-to-last cell in the bottom-right corner of our cartogram map in Figure 1(b). Despite the distortion of the shape of the cell, one can perceive that the data point plotted there is at the center because the connecting line has length zero and so disappears.

Intuitively, all data points should appear somewhere within the grid cell representing their best-match node. If a map has very high quantization errors, data points might be pushed into adjacent grid cells. The connecting line makes this immediately evident; without it, the data points might be seen as belonging to the wrong cells.

4. Examples

For our first experiment we selected the very-well-known Fisher/Anderson "Iris" data [5][6] to demonstrate this visualization. This dataset is included in R's built-in datasets package. Plot symbols are assigned to three iris species: square=*setosa*; circle=*versicolor*; and triangle=*virginica*.

4.1. Cluster Population Size

Each of the three iris species is observed 50 times. In Figure 1, we see that the *versicolor* species appears in 28 of the 98 grid cells, *virginica* in 30, and *setosa* in only 23. (19 cells are empty.) In the cartogram representation of the map

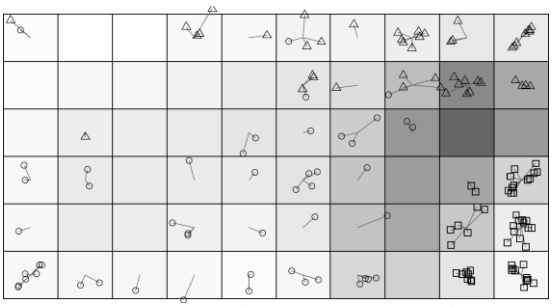


Figure 3. Visualization of the quantization error of a poorly trained SOM.

the cells have been rescaled according to the number of data points mapped to each cell of the SOM.

4.2. Quality and Quantization Error

The average distance in the $\Re^n$ feature space between each training observation and its best-match node measures the overall fit of the map to the data. Maps which minimize this average quantization error are to be preferred [13]. In Figure 3, we have deliberately created a poor quality SOM with high quantization error by limiting the number of training iterations. Despite this, the standard U-Matrix visualization is almost indistinguishable from the map in Figure 1(a). One subtle indication is that there are more empty cells, but that is not very informative. Our visualization in Figure 3 clearly shows the high quantization error with numerous data pushed to and over the edges of their respective cells. In order to emphasize the quantization error we did not apply the cartogram cell expansions in this particular visualization.

5. Cartogram Variations

In Figure 1(b), the cartogram scaling parameter is the number of data points mapped to a cell – the data density of the map. It is possible to use other metrics such as the U-Matrix distance or the number of classes found within a cell as scaling parameters for cartogram visualizations.

In Figure 4, we used the as the U-Matrix distances as a scaling parameter: areas where the U-Matrix distances are low – i.e., areas within clusters [2] – are expanded, area where the U-Matrix distances are high are contracted. This

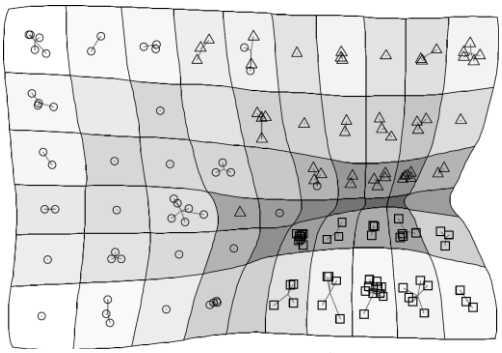


Figure 4. A visualization based on the unified node distance as the cartogram parameter.

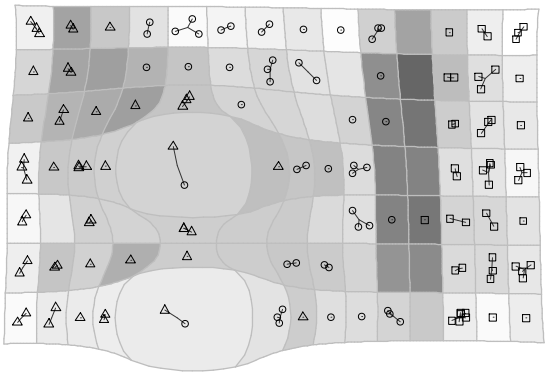


Figure 5. This SOM cartogram emphasizes cells in which more than one species of iris are mapped.

produces an effect similar a contour map or wireframe model where the clusters appear to form "hills" separated by "valleys." While the standard U-Matrix shading is still shown for reference, it is redundant and could be replaced with some other shading such as a color code for class.

The cartogram technique can also be utilized to draw attention to areas of interest, treating the plot as a sort of 3-dimensional pliable surface [15] and pulling the areas of interest "toward" the viewer. The iris data includes three classes (species): one is easily separated, but the other two overlap. In Figure 5, the SOM cells showing this overlap are expanded and so drawn to our attention.

6. Additional Experiments

The iris data set is useful for initial exploration of new techniques, but with only 150 observations and 4 attributes, it is not a good representative of the very large, very-high-dimensional data sets often encountered in real-world settings. The UCI Machine Learning Repository [16] provides a variety of more extensive data sets.

We selected the Cardiotocography [17] data set for further experimentation. A cardiotocogram is a recording of both uterine contractions and fetal heartbeat [18] used in obstetrics. This data set contains results of 2126 examinations each with 21 measured real- or integer-valued attributes plus two additional classification attributes assigned by the consensus of three expert obstetricians. One classification attribute indicates one of ten classes of events

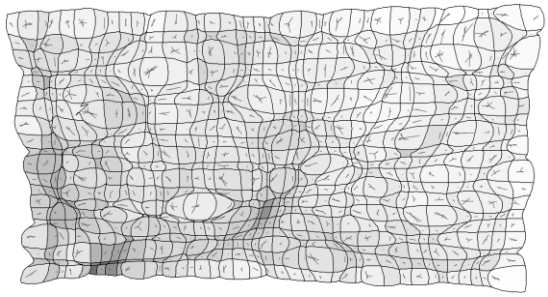


Figure 6. Neither the U-Matrix distances (gray levels) nor data density (cell size) provide insight into the map of cardiotocography data set.

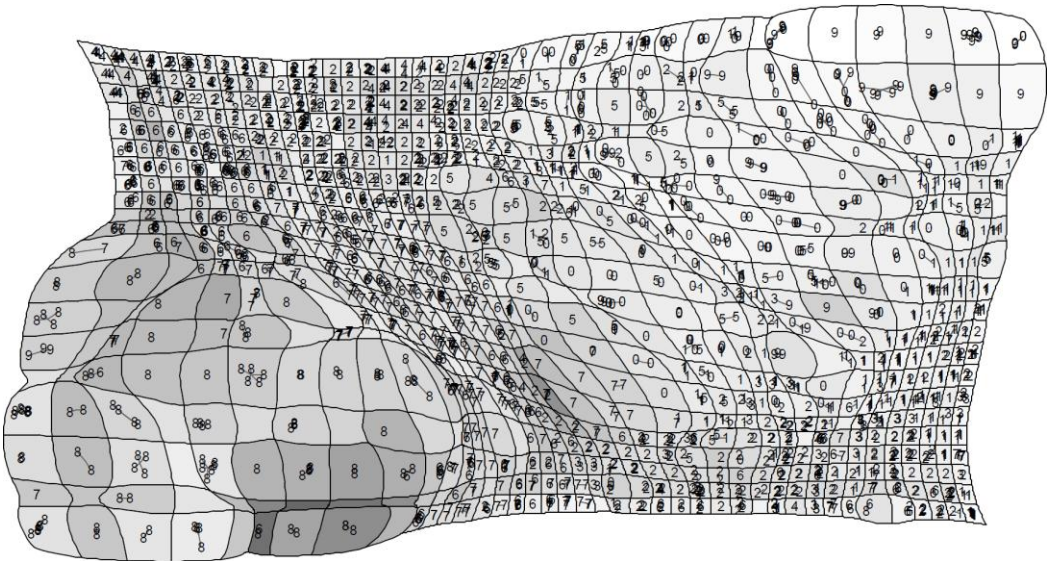


Figure 7. SOM of the cardiotocography data set using the expert assessment of risk as cell size. Numbers are expert-assigned class labels.

being observed such as "calm sleep" or "decelerative pattern." The second classification attribute is a risk measure: "normal," "suspect," or "pathologic."

After normalizing the 26 measured attributes with R's scale function, we constructed a 40×20 rectangular SOM. The expert interpretations (risk and category) were not used to train the SOM. Neither the U-Matrix visualization nor the data density based cartogram of this SOM showed any clear pattern of clustering (Figure 6). There were two pockets of higher U-Matrix distances but no clear divisions between regions. A data density cartogram expansion was not very helpful because the training data are evenly distributed through the map, with cell densities ranging from 0 to 10 with a mean around 2.6 and standard deviation about 1.8.

We obtained a much more interesting plot when we used the cartogram expansion to show the average risk classification (1=normal; 2=suspect; 3=pathologic) of the examination observations mapping to a given cell (Figure 7). This risk assessment was not used to train the SOM, but it seems to correspond to a particular region in the data-space which maps to the lower-left corner of the SOM. The cartogram expansion of that high-risk area gives us more room to see data markers (omitted in Figure 6); the high-risk region is dominated by the codes "largely decelerative" (code 8) and "decelerative" (code 7) of the fetal state classes. As with the risk assessment, the state codes were not used to train the SOM.

Simple color coding would, of course, be able to show the high-risk region, but it would hide the U-Matrix and would not facilitate the closer examination of the data allowed by the cartogram expansion.

7. Related Work

Vesanto [19] demonstrated two methods for showing the quantization error in a SOM. In one, the SOM map is tilted back into a 3-D perspective and bars corresponding to the quantization error project upward. For the second, a circle whose area corresponds to the quantization error surrounds the grid cell. These approaches seem to be appropriate when only a few cells are of interest. They do not lend themselves to seeing the quantization error for the entire map: the bars and circles would obscure each other. Some existing packages will shade the grid cells according to the quantization error (for example, the "quality" map of the 'kohnen' package [20]), but this prevents using shading to show the U-Matrix feature-space distances between cells.

Vesanto [19] also offers methods to visualize the number of data within a cell, potentially helping to understand the size of a cluster. In one, the cell is progressively filled from the center, such that the area of the shape in the center of the cell corresponds to the number of data. The second projects bars upward, making a sort of 3-D histogram. The third "scatters" the data (much like the visualization shown in Figure 1(a)), randomly placing one dot per data sample in the cell. None of these offer any information about the quantization error.

In the Emergent Self-Organizing Map [21][22], Ultsch takes the U*-Matrix – a combination of distance and density – as a height value for the grid. This is shown in 2-D as a sort of topographic contour map that is also colored as in a geographic map (from blue seas to white peaks); the image is also rendered in 3-D. Boundaries are shown as "mountain ranges" separating the "valleys" of the clusters themselves. In our visualization, the cartogram technique, can also present a sort of 3-D appearance as seen in Figure 4. The reshaped edges of the polygons are strongly reminiscent of contour lines on a topographic map.

We have not found any work using a cartogram to aid in the visualization of a SOM, though we did find one paper where the SOM helps to construct a cartogram [23].

8. Implementation in R

8.1. Existing packages

Various packages for training self-organizing maps are available through the Comprehensive R Archive Network (CRAN) [24]. These include 'class' [25], 'RSNNS' [26], 'som' [27] and 'kohonen' [20]. A few other packages offer application-specific visualizations of SOM objects.

Of these, only the 'kohonen' package includes a data mapping visualization. It positions the plotted locations of data points using a randomized normal distribution about the center of the cell, as seen in the "standard" visualization shown in Figure 1(a).

8.2. Our implementation

Our implementation of the cartogram visualization is part of a larger package of SOM visualization and manipulation methods for the R system [28] currently in development. Key features include:

- an S4 adapter class provides a common interface to allow use of SOM objects created by other packages such as 'som'[27] and 'kohonen'[20]
- visualizations use the 'grid' graphics system [29] to facilitate subsequent manipulation and reuse
- support for both square and hexagonal maps
- GPL license

The visualization functions include the capability to construct and display a variety features, including control of:

- cell background shading (i.e., to show the U-Matrix, quantization error, or some other measure),
- individual cell borders,
- the outer borders of contiguous groups of cells (i.e., for outlining clusters),
- the connected components of the map [30] (enhances the display of clusters seen with the U-Matrix),
- data mapping onto the grid and within grid cells, and
- cartogram expansion of the grid.

We anticipate releasing this software through R-forge [31] in mid-2012.

9. Computational complexity

In practice, we find that the calculation of the SOM itself requires far more effort than calculating the data projection within the cell or the cartogram. There are many variations on the basic SOM algorithm; an examination of the implementation in the kohonen package of R shows it to be $O(i \times d \times m \times n)$ where i is the number of iterations over which to train the map; d is the number of observations in the training set; m is the number of nodes in the map; and n is the number of dimensions in the feature space.

At the conclusion of SOM training, the best-match node for each training observation is known and may be saved. Otherwise, finding best-match nodes is a $O(m \times d \times n)$ operation as each node in the map must be examined for each observation. With this information, to locate a single observation within its cell requires a small handful of operations in $\Re^n$ space; it need never consider more than 8 neighbors, so we may say the time required is $O(n)$ for a single observation or $O(d \times n)$ for a set of data.

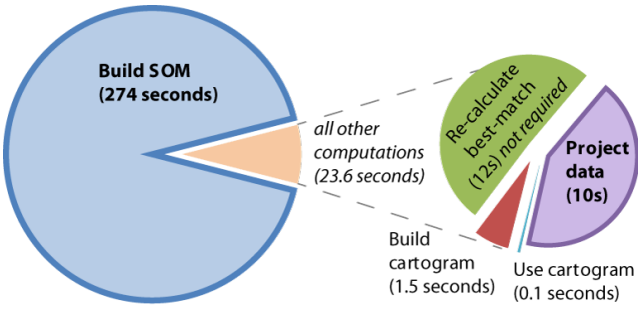


Figure 8. User-time to perform calculations for Figure 7.

Time required process the cartogram is dominated by a Fast Fourier transform: $O(c \log c)$ where c is the number of intersections in the rectangular mesh upon which the cartogram is calculated [7]. We have found that meshes as coarse as 128×128 give acceptable results, though we usually use 256×256. The runtime of the 'sp' package's "overlay" method is a product of the number of points (intersections in the cartogram mesh, c) and the number of polygon edges (a small multiple of the number of nodes in our map). Thus in our use of it, the complexity of "overlay" is $O(c \times m)$.

To quickly confirm our expectations and experience, we used R's system.time() function to display the user-time taken for several of the main computational steps required to produce Figure 7. Figure 8 shows how the SOM calculation itself (274 seconds) far exceeded the time required for other calculations (23.6 seconds). Time to render the illustration itself is not included.

The primary system available for testing and development is a commercially-available notebook computer running 64-bit R 2.13.0 under Windows 7. This machine's CPU (Intel Core i7-2820QM @ 2.30GHz) has four cores, but no computation was ever observed to use more than a single core. R's memory use peaked under 200MB, a very modest footprint. Time tests were repeated at least twice and did not vary by more than one percent of the reported value despite leaving numerous other applications running on the system.

10. Conclusion and Future Work

With this visualization, we have overcome two limitations of the standard SOM visualization: (a) data population visualization for clusters and (b) visualization of the quantization error of a map. Our application of the density diffusion cartogram to the U-Matrix scales clusters in proportion to their population. Information about the quantization error and structure of data points mapped to individual cells is revealed by positioning each point according to its similarity to neighbors.

Some further work is warranted to refine the method of scaling the 2-D data offsets. Furthermore, support for toroidal SOMs (where the edges of the maps are joined) in

the data position calculations would extend our cartogram technique to a few more applications of the SOM.

The cartogram's adaptability to represent any of a variety of different aspects of the data is quite intriguing. It nicely complements shading/coloring and labeling, adding another layer of information to the SOM visualization without overwhelming our ability to understand the image. We intend to investigate additional ways to use this capability in data mining and exploration.

11. References

[1] T. Kohonen, *Self-Organizing Maps*, 3rd ed. Berlin, Heidelberg, New York: Springer, 2001.

[2] A. Ultsch, "Self-Organizing Neural Networks for Visualisation and Classification," in *Information and classification: concepts, methods, and applications*, University of Dortmund, 1993, pp. 307-313.

[3] T. Kohonen, J. Hynninen, J. Kangas, and J. Laaksonen, "SOM PAK: The self-organizing map program package," *Report A31, Helsinki University of Technology, Laboratory of Computer and Information Science*, 1996.

[4] M. Newman, "Images of the social and economic world." [Online]. Available: http://www-personal.umich.edu/~mejn/cartograms/. [Accessed: 26-Feb-2012].

[5] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Human Genetics*, vol. 7, no. 2, pp. 179-188, 1936.

[6] E. Anderson, "The irises of the Gaspe Peninsula," *Bulletin of the American Iris society*, no. 59, pp. 2-5, 1935.

[7] M. T. Gastner and M. E. J. Newman, "Diffusion-based method for producing density-equalizing maps," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 20, pp. 7499-7504, May 2004.

[8] M. T. Gastner and M. E. J. Newman, "Supporting Text for: Diffusion-based method for producing density-equalizing maps." [Online]. Available: http://www.pnas.org/content/101/20/7499/suppl/DC1. [Accessed: 17-Jun-2011].

[9] E. J. Pebesma and R.S. Bivand, "Classes and methods for spatial data in R," *R News*, vol. 5, no. 2, 2005.

[10] T. Zumbrunn, "R-Forge: Diffusion-based cartograms," 26-Sep-2010. [Online]. Available: https://r-forge.r-project.org/projects/cart/. [Accessed: 01-Dec-2010].

[11] D. Temple Lang, "Rcartogram: Interface to Mark Newman's cartogram software," 15-Nov-2008. [Online]. Available: http://www.omegahat.org/Rcartogram/. [Accessed: 16-Jun-2011].

[12] Newman, M. E. J., "Cart: Computer software for making cartograms," 09-Nov-2006. [Online]. Available: http://www-personal.umich.edu/~mejn/cart/. [Accessed: 17-Jun-2011].

[13] T. Kohonen, J. Hynninen, J. Kangas, and J. Laaksonen, "SOM PAK: The Self-Organizing Map Program Package," Helsinki University of Technology, Laboratory of Computer and Information Science, FIN-02150 Espoo, Finland, Technical Report A31, 1996.

[14] D. C. Lay, *Linear Algebra and Its Applications, 3rd Updated Edition*, 3rd ed. Addison Wesley, 2005.

[15] M. S. T. Carpendale, D. J. Cowperthwaite, and F. D. Fracchia, "3-dimensional pliable surfaces: for the effective presentation of visual information," *Proceedings of the 8th annual ACM symposium on User interface and software technology*, pp. 217–226, 1995.

[16] A. Frank and A. Asuncion, "UCI Machine Learning Repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml. [Accessed: 18-Feb-2011].

[17] D. Ayres-de Campos, J. Bernardes, A. Garrido, J. Marques-de-Sa, and L. Pereira-Leite, "SisPorto 2.0: a program for automated analysis of cardiotocograms.," *J Matern Fetal Med*, vol. 9, no. 5, pp. 311-8, 2000.

[18] "Cardiotocography - Wikipedia, the free encyclopedia." [Online]. Available: http://en.wikipedia.org/wiki/Cardiotocography. [Accessed: 27-Feb-2012].

[19] J. Vesanto, "SOM-based data visualization methods," *Intelligent Data Analysis*, vol. 3, no. 2, pp. 111-126, Aug. 1999.

[20] R. Wehrens and L. M. C. Buydens, "Self- and Super-organising Maps in R: the kohonen package," *J. Stat. Softw.*, vol. 21, no. 5, 2007.

[21] A. Ultsch and F. Mörchen, "ESOM-Maps: tools for clustering, visualization, and classification with Emergent SOM," Dept. of Mathematics and Computer Science, University of Marburg, Germany, Technical Report No. 46, 2005.

[22] A. Ultsch, *U*-matrix: a tool to visualize clusters in high dimensional data*. Fachbereich Mathematik und Informatik, 2003.

[23] R. Henriques, F. BaCao, and V. Lobo, "Carto-SOM: cartogram creation using self-organizing maps," *Int. J. Geogr. Inf. Sci.*, vol. 23, no. 4, pp. 483-511, 2009.

[24] T. R Foundation for Statistical Computing, "The Comprehensive R Archive Network." [Online]. Available: http://cran.r-project.org/. [Accessed: 18-May-2011].

[25] W. N. Venables and B. D. Ripley, *Modern Applied Statistics with S*, Fourth. New York: Springer, 2002.

[26] C. Bergmeir and J. M. Benítez, *Neural Networks in R using the Stuttgart Neural Network Simulator: RSNNS*. 2010.

[27] J. Yan, "som: Self-Organizing Map," 2010. [Online]. Available: http://CRAN.R-project.org/package=som. [Accessed: 16-Jun-2011].

[28] R Development Core Team, *R: A Language and Environment for Statistical Computing*. Vienna, Austria: , 2011.

[29] P. Murrell, *R Graphics*, 1st ed. Chapman and Hall/CRC, 2005.

[30] L. Hamel and C. Brown, "Improved interpretability of the Unified Distance Matrix with Connected Components," in *Proceeding of the 7th International Conference on Data Mining*, Las Vegas Nevada, USA, 2011, pp. 338-343.

[31] "R-Forge: Welcome." [Online]. Available: https://r-forge.r-project.org/. [Accessed: 25-May-2011].

A Population Based Convergence Criterion for Self-Organizing Maps

Lutz Hamel and Benjamin Ott
Department of Computer Science and Statistics,
University of Rhode Island,
Kingston, RI 02881, USA.

Abstract— *Self-organizing maps are a type of artificial neural network extensively used as a data mining and analysis tool in a broad variety of fields including bioinformatics, financial analysis, signal processing, and experimental physics. They are attractive because they provide a simple yet effective algorithm for data clustering and visualization via unsupervised learning. A fundamental question regarding self-organizing maps is the question of convergence or how well the map models the data after training. Here we introduce a population based convergence criterion: the neurons of the map represent one population and the training data represents another population. The map is said to be converged if the neuron and the training data populations appear to be drawn from the same probability distribution. This can easily be tested with standard two-sample tests. This paper develops the underpinnings of this approach and then applies this new convergence criterion to real-world data sets. We demonstrate that our convergence criterion can be considered an appropriate model selection criterion.*

Keywords: self-organizing map, convergence, quantization error, two-sample test

1. Introduction

Self-organizing maps (SOMs) are a type of artificial neural network extensively used as a data mining and analysis tool in a broad variety of fields including bioinformatics, financial analysis, signal processing, and experimental physics [9]. The straight forward nature of the SOM training algorithm and the way in which the visualization of a SOM can be easily and intuitively interpreted make it appealing as an analysis tool. However, as with any analysis tool, and especially with competitive learning-based tools, questions pertaining to the reliability and the convergence of the tool naturally emerge. Here we view convergence as a measure of how well a model represents the underlying data space. In SOMs, convergence, and therefore the quality of the produced visualization, critically depends on the number of neurons selected and the number of training iterations applied to those neurons. If we view SOMs as models of the training data then any convergence criterion essentially becomes a model selection criterion allowing us to distinguish "good models" from "poor models."

Several measures have been developed in order to analyze the convergence of a given SOM. One such measure, the quantization error, is the error function proposed by Kohonen and is the *de facto standard* measure of the convergence of a given SOM [9]. The quantization error of a given training observation is the smallest distance between that training observation and any neuron in the SOM. The quantization error of a training set is typically the mean sum squared quantization error of all training instances. The goal of the SOM algorithm then, is to minimize this value during training. Attempting to minimize the quantization error in a radical fashion (minimize it to zero, for example) can lead to overfitted models which may be ineffective at representing the data at hand because they may end up modelling noise in the training data which is not characteristic of the general population from which the training data sample was drawn. Since one can make the quantization error arbitrarily small by increasing the complexity of the model by adding neurons to the map and by increasing the training iterations, it is clear that the quantization error does not lend itself as a model selection criterion, since it cannot answer the question "When is the quantization error good enough?"

Another approach to obtaining measurable convergence criteria is to modify the SOM training algorithm itself so that statistical measures or other objective analysis techniques can be imposed. Bishop's generative topographic mapping (GTM) [3] and Verbeek's generative self-organizing map (GSOM) [11] seem to fall into this category. The GTM and GSOM attempt to model the probability density of a data set using a smaller number of latent variables (i.e. the dimensionality of the latent space is less than or equal to that of the data space). A non-linear mapping is then generated which maps the latent space into the data space. The parameters of this mapping are learned using using an expectation-maximization (EM) algorithm. Algorithms such as the GTM and the GSOM should be viewed as alternates to the SOM as opposed to modifications of it, even though they share properties similar to the SOM. Other scholars have taken an energy function approach, imposing energy functions on the SOM and then attempting to minimize

these energy functions [8, 6]. Both of these approaches, namely altering the algorithm or imposing energy functions on the SOM, seem to take away from the SOM's appeal as a simple, fast algorithm for visualizing high dimensional data, especially since the alterations tend to be much more complex than the SOM learning algorithm itself.

Yet another approach is to calculate the significance of neighborhood stabilities in order to analyze whether or not data points close together in the input space remain close when projected onto the SOM. By analyzing many maps which were trained with bootstrap samples drawn from the training data, this approach by Cottrell *et al* [4] provides a sound set of statistical tools to analyze SOMs while leaving the original SOM algorithm unchanged. However, this stability based approach is computationally unwieldy drastically increasing the amount of time associated with the analysis of a given data set. The increased time cost is due to the creation of many maps using bootstrapped samples of the training data (typically 100-200 maps; Efron recommends using at least 200 samples when bootstrapping statistics [5]) and the analysis of each pair of training data over each map after all of the maps have been created.

In this paper, we propose a population based approach for analyzing SOM convergence. Under some minor simplifying assumptions, Yin and Allison [12] showed that, in the limit, the neurons of a SOM will converge on the probability distribution of the training data. This seems to validate Kohonen's claim that a SOM will in effect model the probability distribution of the training data [9]. Hence, a simple two-sample test can be performed in order to see if the SOM has effectively modelled the probability distribution formed by the training data or not. This population based approach lends to a fast convergence criterion, based on standard statistical methods, which does not modify the original algorithm, hence preserving its appeal as a simple and fast analysis tool.

The remainder of this paper is structured as follows. Section 2 describes the basic SOM training algorithm. In Section 3 we investigate convergence properties of this algorithm in the limit following [12]. We look at model selection and quantization error in Section 4 and we introduce our population based convergence criterion in Section 5. We illustrate our convergence criterion with examples in Sections 6 and 7. Finally, we state our conclusions and further work in Section 8.

2. The Basic Algorithm

Here we describe the training algorithm for SOM as introduced by Kohonen [9] following the notation used by Yin and Allison [12]. The canonical training algorithm for SOM uses a set of neurons, $\mathbf{Y}$, usually arranged in a rectangular grid formation to form topology preserving discrete mappings of an N-dimensional input space $\mathbf{X} \subset \mathbb{R}^N$. Each element $\mathbf{x} \in \mathbf{X}$ is considered a training instance and is a vector $\mathbf{x} = [x_1, x_2, \ldots, x_N]^T$ and each neuron indexed by $c \in \mathbf{Y}$ is a vector $\mathbf{w}_c(t) \in \mathbb{R}^N$ with

$$\mathbf{w}_c(t) = [w_{c,1}(t), w_{c,2}(t), \ldots, w_{c,N}(t)]^T, \tag{1}$$

where t is a time step index. Each neuron is a weight vector of the same dimensionality as the input space and the weights are adjusted at each discrete time step t during training with $t \geq 0$. At each time step t a randomly selected input vector $\mathbf{x}(t) \in \mathbf{X}$ is chosen and is used to compute a winning neuron $\omega(t)$,

$$\omega(t) = \arg\min_{c \in \mathbf{Y}} \parallel \mathbf{x}(t) - \mathbf{w}_c(t) \parallel . \tag{2}$$

Here the winning neuron $\omega(t)$ has the smallest Euclidean distance $\parallel \mathbf{x}(t) - \omega(t) \parallel$ from the training instance $\mathbf{x}(t)$. Once the winning neuron has been found the weights of the neurons on the map are updated according to the following rule,

$$\mathbf{w}_c(t+1) = \mathbf{w}_c(t) + \alpha(t)\, h_{c,\omega(t)}(t)\, [\mathbf{x}(t) - \mathbf{w}_c(t)], \tag{3}$$

for all $c \in \mathbf{Y}$. Here $\alpha(t)$ is the *learning rate* at time step t (typically a small constant with $0 < \alpha(t) < 1$ for all t decaying as t grows large) and $h_{c,\omega(t)}(t)$ is the *neighborhood function* at time step t indexed by c and the winning neuron $\omega(t)$. The neighborhood of a winning neuron, $\mathbf{N}_{\omega(t)}(t)$, is defined as a set of neurons in proximity of and centered around the winning neuron according to the grid formation of the map. The neighborhood is time step sensitive and usually starts out as a large set,

$$\mathbf{N}_{\omega(t)}(t) \approx \mathbf{Y}, \tag{4}$$

at $t = 0$ and shrinks over time,

$$\mathbf{N}_{\omega(t)}(t) = \{\omega(t)\}, \tag{5}$$

with $t \gg 0$. With this we define the neighborhood function as the step function,

$$h_{c,\omega(t)}(t) = \begin{cases} 1 \text{ if } c \in \mathbf{N}_{\omega(t)}(t) \\ 0 \text{ otherwise} \end{cases} \tag{6}$$

This implies that the neighborhood function controls which neurons on the map are updated according to the rule in (3) and which are not. This "focussing" of neuron updating is crucial in the successful training of SOMs. Training usually continues for a fixed number of time steps.

3. Probabilistic Convergence

Some observations on the training of a SOM. Training instances are drawn from the input space $\mathbf{X}$ randomly and independently and are presented to the map; at time step t we can view the randomly chosen training instances as the set,

$$\mathbf{X}(t) = \{\mathbf{x}(i) \in \mathbf{X} \mid i = 0, \ldots, t\}, \tag{7}$$

with $\mathbf{X}(t) \overset{t\to\infty}{\longrightarrow} \mathbf{X}$. At time step t, each neuron $c \in \mathbf{Y}$ is trained with a subset of instances $\mathbf{X}_c(t) \subseteq \mathbf{X}(t)$ where,

$$\bigcup_{c\in\mathbf{Y}} \mathbf{X}_c(t) = \mathbf{X}(t). \tag{8}$$

At the beginning of the training process, when the neighborhoods are still large, i.e., $\mathbf{N}_{\omega(t)}(t) \approx \mathbf{Y}$, these subsets are maximally overlapped with each other. As training progresses and the neighborhoods shrink to a single element, i.e., $\mathbf{N}_{\omega(t)}(t) = \{\omega(t)\}$, these subsets eventually become mutually separated with,

$$\mathbf{X}_c(t) \bigcap \mathbf{X}_{c'}(t) \overset{t\to\infty}{\longrightarrow} \emptyset, \tag{9}$$

for all $c, c' \in \mathbf{Y}$ and $c \neq c'$. Furthermore, as time tends to infinity we have,

$$\mathbf{X}_c(t) \overset{t\to\infty}{\longrightarrow} \mathbf{X}_c, \tag{10}$$

the *final* subsets.

Now, let $p(\mathbf{x})$ be the probability density function over the input space $\mathbf{X}$, then the probability of a training instance $\mathbf{x}(t)$ belonging to a subset $\mathbf{X}_c(t)$ is,

$$P(\mathbf{X}_c(t)) = \int_{\mathbf{x}\in\mathbf{X}_c(t)} p(\mathbf{x})d\mathbf{x}, \tag{11}$$

for all $c \in \mathbf{Y}$. As $t \to \infty$ this becomes,

$$P(\mathbf{X}_c) = \int_{\mathbf{x}\in\mathbf{X}_c} p(\mathbf{x})d\mathbf{x}. \tag{12}$$

Yin and Allison [12] have shown under some mild assumptions that for a given map, $\mathbf{Y}$, the neurons will converge in the limit on the centroids $\mathbf{m}_c$ of the final subsets $\mathbf{X}_c$,

$$\mathbf{w}_c(t) \overset{t\to\infty}{\longrightarrow} \mathbf{m}_c = \frac{1}{P(\mathbf{X}_c)} \int_{\mathbf{x}\in\mathbf{X}_c} \mathbf{x}\, p(\mathbf{x})d\mathbf{x}, \tag{13}$$

for all $c \in \mathbf{Y}$.

4. Model Selection and Quantization Error

We define the *quantization error* for a map $\mathbf{Y}$ at time step t as,

$$E_{\mathbf{Y}}(t) = \sum_{c\in\mathbf{Y}} \sum_{\mathbf{x}\in\mathbf{X}_c(t)} \parallel \mathbf{w}_c(t) - \mathbf{x} \parallel^2, \tag{14}$$

and it is easy to see given (13) that the learning algorithm converges on the minimal quantization error as time tends to infinity,

$$E_{\mathbf{Y}}(t) \overset{t\to\infty}{\longrightarrow} E_{\mathbf{Y}} = \sum_{c\in\mathbf{Y}} \sum_{\mathbf{x}\in\mathbf{X}_c} \parallel \mathbf{m}_c - \mathbf{x} \parallel^2 . \tag{15}$$

If we view map construction as a model building process and if we view the quantization error as a model selection criterion (as suggested in [9]) we run into problems in that optimality is defined only in the limit. There is no statistical measure on the quantization error that suggests when a quantization error is "good enough." To complicate things even further, adding neurons to the map and rerunning the training algorithm will likely reduce the quantization error because now the algorithm will split the training set into a larger number of final subsets (one per neuron) with fewer training instances in them which will give rise to a lowered quantization error. Again, no statistical measure based on the quantization error exists that would suggest an optimal number of nodes.

Given what we have developed so far and assuming that the topology of the input space $\mathbf{X}$ can be projected onto two dimensions with minimal distortion, then it is possible to reduce the quantization error to zero by constructing a large enough map. To see this, let n_x be the number of elements in the input space $\mathbf{X}$ and let n_y be the number of neurons in the map $\mathbf{Y}$. First assume that the map only consists of a single neuron, $n_y = 1$. That means in the limit we have $P(\mathbf{X}_c) = P(\mathbf{X}) = 1$ and therefore the single neuron will converge on the mean $\mathbf{m}_x$ of the input space,

$$\mathbf{w}_c(t) \overset{t\to\infty}{\longrightarrow} \mathbf{m}_x = \int_{\mathbf{x}\in\mathbf{X}} \mathbf{x}\, p(\mathbf{x})d\mathbf{x}, \tag{16}$$

for the only element $c \in \mathbf{Y}$. This implies of course a large quantization error,

$$E_{\mathbf{Y}}(t) \overset{t\to\infty}{\longrightarrow} E_{\mathbf{Y}} = \sum_{\mathbf{x}\in\mathbf{X}} \parallel \mathbf{m}_x - \mathbf{x} \parallel^2 . \tag{17}$$

Now assuming that we have as many neurons in our map as there are training instances, $n_y = n_x$, and making use of our assumption that the topology of the input space can be projected onto two dimensions with minimal distortion, then our final subsets could be singleton sets $\mathbf{X}_c = \{\mathbf{x}_c\}$ for all $c \in \mathbf{Y}$where,

$$\bigcup_{c\in\mathbf{Y}} \mathbf{X}_c = \mathbf{X}, \tag{18}$$

and

$$\bigcap_{c\in\mathbf{Y}} \mathbf{X}_c = \emptyset. \tag{19}$$

In the limit, each neuron of the map will then converge on the training instance $\mathbf{x}_c$ in its final subset,

$$\mathbf{w}_c(t) \overset{t\to\infty}{\longrightarrow} \mathbf{x}_c = \frac{1}{P(\{\mathbf{x}_c\})} \int_{\{\mathbf{x}_c\}} \mathbf{x}\, p(\mathbf{x})d\mathbf{x}, \tag{20}$$

for all $c \in \mathbf{Y}$. It is easy to see that the quantization error in the limit will be zero in this case. This means that quantization error as a model selection criterion dictates that the neurons of the SOM have to model the training data precisely. But statistical theory tells us that models that fit their training data precisely are usually overfitted since, by modeling training data precisely, these models also model any inherent noise.

5. Population Based Convergence

As we have seen in the previous section, the quantization error is not an adequate model selection criterion because we can make the errors as small as desired by increasing the complexity of the model. It is required of a model selection criterion that it allows us to determine when a model is "good enough" which the quantization error does not allow us to determine.

However, by slightly shifting perspective, another look at equation (20) reveals something interesting about SOMs: given enough neurons the SOM training algorithm attempts to recreate the training samples. This insight allows us to formulate a new convergence criterion:

> *A SOM is converged if its neurons appear to be drawn from the same distribution as the training instances.*

This formulation naturally leads to a two-sample test [10] as a convergence criterion. We can view the training data as one sample from the probability space $\mathbf{X}$ having probability density function $p(x)$ and we can treat the neurons of the SOM as another sample. We can then test to see whether or not the two samples appear to be drawn from the same probability space. If we operate under the simplifying assumption that each of the N coordinates (or features) of the input space $\mathbf{X} \subset \mathbb{R}^N$ are normally distributed and independent of the others, we can test each of the features separately. This assumption lends to a fast algorithm for identifying SOM convergence which is based on statistically analyzing similarites between the features as expressed by the training data and as expressed by the neurons in the SOM. We define a feature as converged if the variance and the mean of that feature appear to be drawn from the same distribution for both the training data and the neurons. If all the features are converged then we say that the map is converged.

The following is the formula for the $(1-\alpha) * 100\%$ confidence interval for the ratio of the variances from two random samples [10],

$$\frac{s_1^2}{{s_2}^2} \cdot \frac{1}{f_{\frac{\alpha}{2},n_1-1,n_2-1}} < \frac{\sigma_1^2}{\sigma_2^2} < \frac{s_1^2}{{s_2}^2} \cdot f_{\frac{\alpha}{2},n_1-1,n_2-1}, \quad (21)$$

where s_1^2 and s_2^2 are the values of the variance from two random samples of sizes n_1 and n_2 respectively, and where $f_{\frac{\alpha}{2},n_1-1,n_2-1}$ is an F distribution with n_1-1 and n_2-1 degrees of freedom. To test for SOM convergence, we let s_1^2 be the variance of a feature in the training data and we let s_2^2 be the variance of that feature in the neurons of the map. Furthermore, n_1 is the number of training samples (i.e. the cardinality of the training data set) and n_2 is the number of neurons in the SOM. We say that the variance of a particular feature has converged (or appears to be drawn from the same probability space) if 1 lies in the confidence interval denoted by equation (21), that is, the ratio of the underlying variance as modeled by input space and the neuron space, respectively, is approximately equal to one, $\sigma_1^2/\sigma_2^2 \approx 1$, up to the confidence interval.

In the case where $\bar{x}_1$ and $\bar{x}_2$ are the values of the means from two random samples of size n_1 and n_2, and the known variances of these samples are σ_1^2 and σ_2^2 respectively, the following formula provides $(1-\alpha) * 100\%$ confidence interval for the difference between the means [10],

$$\mu_1 - \mu_2 > (\bar{x}_1 - \bar{x}_2) - z_{\frac{\alpha}{2}} \cdot \sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}, \quad (22)$$

$$\mu_1 - \mu_2 < (\bar{x}_1 - \bar{x}_2) + z_{\frac{\alpha}{2}} \cdot \sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}. \quad (23)$$

We will say that the mean of a particular feature has converged if 0 lies in the confidence interval denoted by equations (22) and (23). That is, we say the mean of a particular feature has converged if the difference of the means as modeled by the input space and the neuron space, respectively, is approximately equal to zero, $\mu_1 - \mu_2 \approx 0$, up to the confidence interval.

We will say that a SOM has converged on a feature, or that a feature has converged, if both the mean and variance converged in accordance with the above criteria. We can then form a measure for SOM convergence as follows for N features,

$$\textit{convergence} = \frac{\sum_{i=1}^{N} \rho_i}{N}, \quad (24)$$

where

$$\rho_i = \begin{cases} 1 & \text{if feature } i \text{ has converged,} \\ 0 & \text{otherwise.} \end{cases}$$

The convergence score (24) proposed here is essentially a fraction of the number of features which actually converged (i.e. whose mean and variance were adequately modelled by the neurons in the SOM).

6. Example Using Iris Data

The convergence measure proposed in the previous section was applied to the Fisher / Anderson iris data set [1] using 95% confidence intervals for both the mean and the variance tests. In the following plots fConv represents the convergence measure as defined in the previous sections, ssMean is the quantization error as defined in (14), and iterations represents the number of iterations that the SOM training algorithm was run. Each data point represents the data associated with a unique randomly initialized SOM which was trained for the number of iterations indicated on the plot. For instance, when fConv is plotted against iterations (Figure 1), the data points on the map, which are connected via lines for visualization purposes, represent the

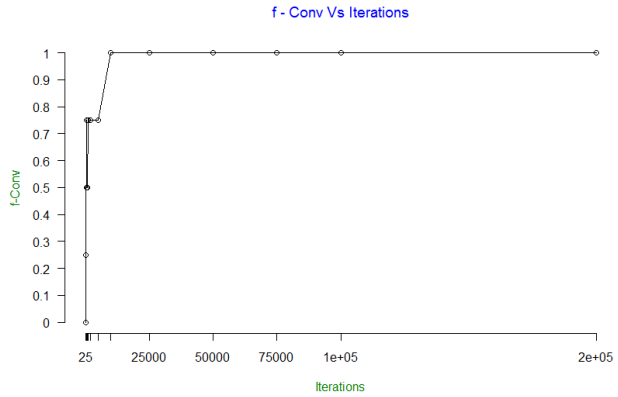


Fig. 1: Convergence measure, fConv, plotted as a function of iterations.

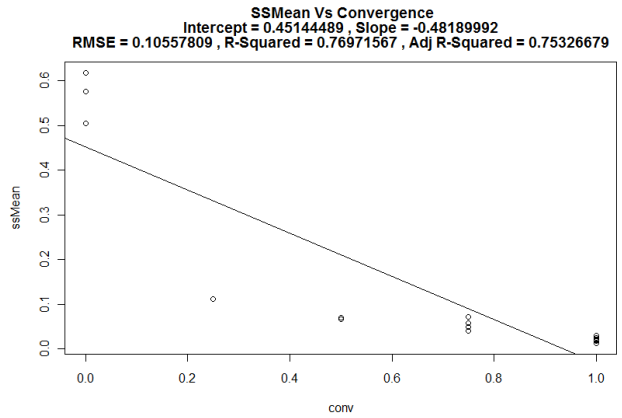


Fig. 2: ssMean plotted as a linear function of fConv.

convergence score for unique, randomly initialized SOMs which were trained for the indicated number of iterations.

In Figure 1, we can see that the our convergence measure increases as the amount of training increases. This is expected for any valid convergence criterion. We can also see that SOM has fully converged after about 20,000 iterations. Figure 2 shows the convergence measure related to the quantization error. We can see that as the convergence score increases, the quantization error decreases. Most interesting is the fact that fully converged maps (points at the bottom right) do not necessarily have a quantization error of zero making our convergence score a suitable model selection criterion as the neurons in these converged maps adequately model the input data without overfitting. This also implies that if the convergence score falls substantially short of the value 1, then we can improve the models by either training longer, adding neurons, increasing the learning rate, or all of the above. Each of these steps enables the SOM to more easily capture the variance of the training data, thereby increasing the possibility of convergence.

7. Example Using Wine Data

In this section, our convergence measure was applied to Stefan Aeberhard's wine data set [2] using 95% confidence intervals for both the mean and the variance tests. The data was normalized before training the SOM. This data has thirteen dimensions and consists of chemical characteristics for three types of wine and we would expect three clusters to be shown on converged maps. Again as above, in the following plots fConv represents the convergence measure as defined in the previous sections, ssMean is the quantization error as defined in (14), and iterations represents the number of iterations that the SOM algorithm was run. Each data point represents the data associated with a unique randomly initialized SOM which was trained for the number of iterations indicated by the data point.

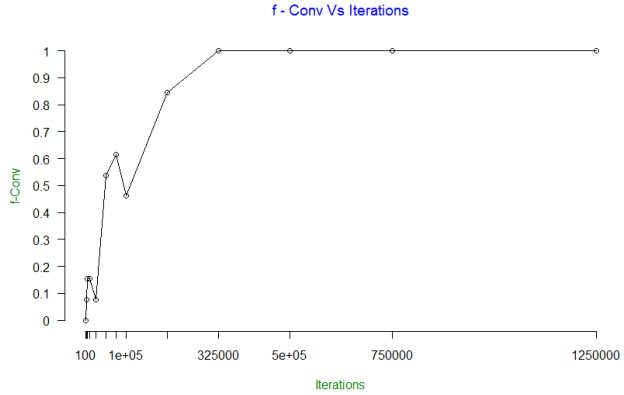


Fig. 3: Convergence measure, fConv, plotted as a function of iterations.

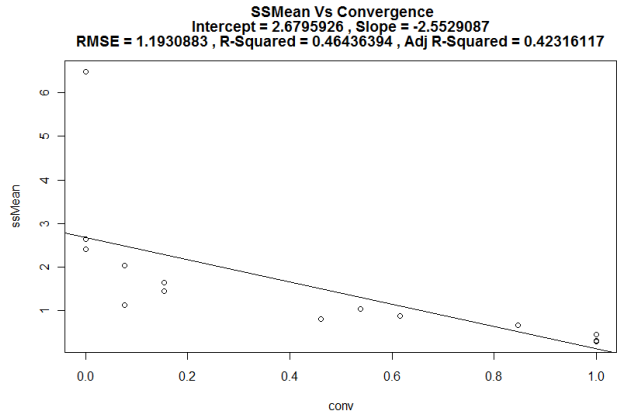


Fig. 4: ssMean plotted as a linear function of fConv.

In Figure 3, we can see that, as expected, our convergence score trends upwards as the amount of training increases. Note also that this occurs even though each of the maps are randomly initialized. We can also see that maps fully

converge after about 325,000 iterations. Figure 4 shows us that the convergence measure is inversely related to the quantization error. We can see that as convergence increases, the quantization error decreases. As before, fully converged maps (points at the bottom right) do not necessarily have a quantization error of zero. The neurons in these maps adequately model the input data without overfitting.

In Figures 5 and 6 we present two SOMs constructed using the wine data set, one with a low convergence score and one with a high convergence score, respectively.

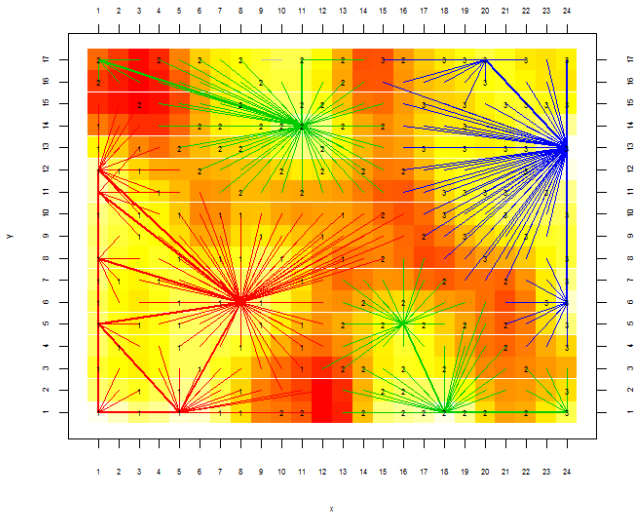


Fig. 5: Map trained using the wine data set for 5,000 iterations achieving a convergence score of 15.8%.

The map in Figure 5 has a convergence score of about 15% after 5,000 training iterations and one can easily see from the map that the cluster with elements '2' was broken into two parts: one part can be seen in the top left corner and the other in the bottom right. This is contrary to our expectation of being able to identify three contiguous clusters. In Figure 6 the map achieved a 100% score with 500,000 training iterations and produced the expected clusters. Here we can identify all three contiguous clusters. The cluster representations were created using a slightly modified version of the connected components approach as given in [7]. As expected, these maps seem to indicate that the quality of a map is directly correlated to its convergence score: the higher the convergence score, the better the map. That means that our convergence score is an appropriate model selection criterion as desired.

8. Conclusion and Further Work

Self-organizing maps are a popular data analysis and visualization tool. However, attempts to provide a convergence criterion for SOMs either resulted in a modified training

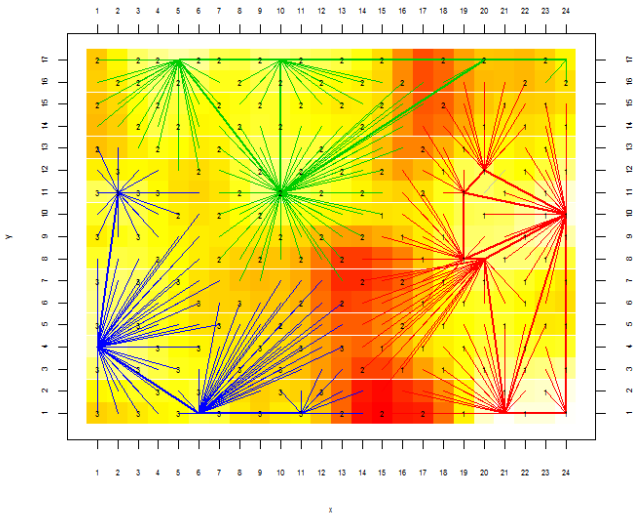


Fig. 6: Map trained using the wine data set for 500,000 iterations achieving a convergence score of 100%.

algorithm or computationally complex constructions. In the case of the quantization error we have shown that it is not suited to be considered a convergence criterion. Here we presented an efficient alternative that treats the neurons as a data sample and convergence of the SOM is established if the neuron sample appears to be drawn from the same distribution as the training data. This two-sample test can be efficiently computed based on the features of the training data. Our examples demonstrated that our convergence criterion is inversely related to the quantization error; convergence increases as quantization error decreases. However, convergence does not necessarily imply a zero quantization error which means that our convergence criterion avoids the overfitting tendencies of quantization error based modeling approaches. Furthermore, our examples seem to indicate that the quality of the maps produced is directly correlated to our convergence score making it an appropriate model selection criterion.

Our next step is to compare our convergence criterion to established convergence criteria such as Cottrell *et al*'s stability measure [4]. Some preliminary studies are encouraging in that our convergence criterion always implies Cottrell's stability criterion. Our goal is to incorporate this new convergence criterion into a comprehensive SOM toolkit R package under development at the University of Rhode Island to be made available to the public.

References

[1] UCI machine learning repository: Iris data set. http://archive.ics.uci.edu/ml/datasets/Iris.

[2] UCI machine learning repository: Wine data set. http://archive.ics.uci.edu/ml/datasets/Wine.

[3] C. Bishop, M. Svensen, and C. Williams. Gtm: A principled alternative to the self-organizing map. *Artificial Neural Networks-ICANN 96*, pages 165–170.

[4] M. Cottrell, E. De Bodt, and M. Verleysen. A statistical tool to assess the reliability of self-organizing maps. *Advances in self-organising maps*, pages 7–14, 2001.

[5] B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, New York, 1993.

[6] E. Erwin, K. Obermayer, and K. Schulten. Self-organizing maps: ordering, convergence properties and energy functions. *Biological cybernetics*, 67(1):47–55, 1992.

[7] L. Hamel and C.W. Brown. Improved interpretability of the unified distance matrix with connected components.

[8] T. Heskes. Energy functions for self-organizing maps. *Kohonen maps*, pages 303–316.

[9] T. Kohonen. *Self-organizing maps*. Springer series in information sciences. Springer, 2001.

[10] Irwin Miller and Marylees Miller. *John E. Freund's Mathematical Statistics with Applications (7th Edition)*. Prentice Hall, 7 edition, October 2003.

[11] J. J. Verbeek and N. Vlassis. The generative self-organizing map.

[12] H. Yin and N. M. Allinson. On the distribution and convergence of feature space in self-organizing maps. *Neural computation*, 7(6):1178–1187, 1995.

Clustering Approaches for Financial Data Analysis: a Survey

Fan Cai, Nhien-An Le-Khac, M-Tahar Kechadi,
School of Computer Science & Informatics, University College Dublin, Ireland

***Abstract*—Nowadays, financial data analysis is becoming increasingly important in the business market. As companies collect more and more data from daily operations, they expect to extract useful knowledge from existing collected data to help make reasonable decisions for new customer requests, e.g. user credit category, confidence of expected return, etc. Banking and financial institutes have applied different data mining techniques to enhance their business performance. Among these techniques, clustering has been considered as a significant method to capture the natural structure of data. However, there are not many studies on clustering approaches for financial data analysis. In this paper, we evaluate different clustering algorithms for analysing different financial datasets varied from time series to transactions. We also discuss the advantages and disadvantages of each method to enhance the understanding of inner structure of financial datasets as well as the capability of each clustering method in this context.**

Keywords-clustering; partitioning clustering; density-based clustering; financial datasets

I. INTRODUCTION

TODAY, we have a deluge of financial datasets. Faster and cheaper storage technology allows us to store ever-greater amounts of data. Due to the large sizes of the data sources it is not possible for a human analyst to come up with interesting information (or patterns) that will help in the decision making process. Global competitions, dynamic markets, and rapid development in the information and communication technologies are some of the major challenges in today's financial industry. For instance, financial institutions are in constant needs for more data analysis, which is becoming more very large and complex. As the amount of data available is constantly increasing, our ability to process it becomes more and more difficult. Efficient discovery of useful knowledge from these datasets is therefore becoming a challenge and a massive economic need.

On the other hand, data mining (DM) is the process of extracting useful, often previously unknown information, so-called knowledge, from large datasets (databases or data). This mined knowledge can be used for various applications such as market analysis, fraud detection, customer retention, etc. Recently, DM has proven to be very effective and profitable in analysing financial datasets [1]. However, mining financial data presents special challenges; complexity, external factors, confidentiality, heterogeneity, and size. The data miners' challenge is to find the trends quickly while they are valid, as well as to recognize the time when the trends are no longer effective. Moreover, designing an appropriate process for discovering valuable knowledge in financial data is a very complex task.

Different DM techniques have been proposed in the literature for data analysing in various financial applications. For instance, decision-tree [2] and first-order learning [3] are used in stock selection. Neural networks [4] and support vector machine [5] techniques were used to predict bankruptcy, nearest-neighbours classification [6] for the fraud detection. Users also have used these techniques for analysing financial time series [7], imputed financial data [8], outlier detection [9], etc. However, there are not many clustering techniques applied in this domain compared to other techniques such as classification and regression [2].

In this paper, we survey different clustering algorithms for analysing different financial datasets for a variety of applications; credit cards fraud detection, investment transactions, stock market, etc. We discuss the advantages and disadvantages of each method in relation to better understanding of inner structure of financial datasets as well as the capability of each clustering method in this context. In other words, the purpose of this research is to provide an overview of how basic clustering methods were applied on financial data analysis.

The rest of this paper is organised as follows. In Section II, we present briefly different financial data mining techniques that can be found in the literature. Section III describes briefly different clustering techniques used in this domain. We evaluate and discuss the advantages and disadvantages of these clustering methods in Section IV. We conclude and discuss some future directions in Section V.

II. DATA MINING IN FINANCE

A. Association Rules

Association Rule is a DM technique known as association analysis, which is useful for discovering interesting relationships hidden in large datasets. These relationships can be represented in the form of association rules or sets of frequent itemsets [2]. This technique can be applied to analyse data in different domains such as finance, earth science, bioinformatics, medical diagnosis, web mining, and scientific computation.

In finance, association analysis is used for instance in

N-A. Le-Khac: School of Computer Science & Informatics, University College Dublin, Ireland (**Corresponding author**: an.lekhac@lucd.ie).
F. Cai: School of Computer Science & Informatics, University College Dublin, Ireland (caifan.home@gmail.com).
M-T. Kechadi: School of Computer Science & Informatics, University College Dublin, Ireland (tahar.kechadi@ucd.ie).

customer profiling that builds profiles of different groups from the company's existing customer database. The information obtained from this process can help understanding business performance, making new marketing initiatives, analysing risks, and revising company customer policies. Moreover, loan payment prediction, customer credit policy analysis, marketing and customer care can also perform association analysis to identify important factors and eliminate irrelevant ones.

B. Classification

Classification is another DM approach, which assigns objects to one of the predefined categories. It uses training examples, such as pairs of input and output targets, to find an appropriate target function also known informally as a classification model. The classification model is useful for both descriptive and predictive modelling [2]. In finance, classification approaches are also used in customer profiling by building predictive models where predicted values are categorical. Financial market risk, credit scoring/rating, portfolio management, and trading also apply this approach to group similar data together.

Classification can be considered as one of the important analytical methods in computational finance. Rule-based methods [2][3] can be used for the stock selection. Besides, bankruptcy prediction can use its geometric methods [4][5] where classification functions are represented with a set of decision boundaries constructed by optimising certain error criteria. Other methods such as Naïve Bayes classifiers [10], maximum entropy classifiers [11] were applied in bond rating and prototype-based classification methods such as nearest-neighbours classification was moreover used for the fraud detection.

C. Clustering

Like classification, cluster analysis groups similar data objects into clusters [2], however, the classes or clusters were not defined in advance. Normally, clustering analysis is a useful starting point for other purposes such as data summarisation. A cluster of data objects can be considered as a form of data compression. Different domains can apply clustering techniques to analysis data such as biology, information retrieval, medicine, etc. In the business and finance, clustering can be used, for instance, to segment customers into a number of groups for additional analysis and marketing activities. As clustering is normally used in data summarisation or compression, there are not many financial applications that use this technique compared to classification and association analysis. We will survey some approaches in Section III.

D. Other methods

Other mining techniques that can be applied for financial datasets are grouped in three categories: optimization, regression and simulation. For instance, portfolio selection, risk management and asset liability management can use different optimisation techniques such as genetic algorithms [12], dynamic programming [13], reinforcement learning [14], etc. Besides, linear regression [2] and wavelet regression [15] are popular methods in the domain of financial forecasting, option pricing and stock prediction.

III. CLUSTERING METHODS

A. Partitioning Methods

K-means clustering [16] method aims to partition **n** observed examples into **k** clusters. Each example belongs to one cluster. All examples are treated with the equal importance and thus a mean is taken as the centroid of the observations in the cluster. With the predetermined **k**, the algorithm proceeds by alternating between two steps: assignment step and update step. Assignment step assigns each example to its closest cluster (centroid). Update step uses the result of assignment step to calculate the new means (centroids) of newly formed clusters. The convergence speed of the k-means algorithm is fast in practice but the optimal **k** value is not known in advance.

In [17], the author uses k-means algorithm to categorise mutual funds. The created clusters are assigned according to self-declared investment objectives and are compared to explain the difference between expectation and financial characteristics. Besides, in order to determine the number of clusters (k), the author applied the Hartigan's theory by evaluating the following formula:

$$\left(\frac{\sum_{i=1}^{k} ESS}{\sum_{i=1}^{k+1} ESS} - 1\right) \times (n - k - 1) > 10 \qquad (1)$$

where k is the result with k clusters and ESS represents the sum of squares and n is the dataset's size. The number of clusters is the minimum *k* such that (1) is false.

B. Density-based

Another clustering approach is density based [2] which does not partition the sample space by mean centroid, but instead density based information is used, by which tangled, irregular contoured but well distributed dataset can be clustered correctly.

OPTICS [18] is a density based clustering technique to get insight into the density distribution of a dataset. It makes up for the weakness of the k-means algorithm for lack of knowledge of how to choose the value k. OPTICS provides a perspective to look into the size of density-based clusters.

Unlike centroid-based clustering, OPTICS does not produce a clustering of a dataset explicitly from the first step. It instead creates an augmented ordering of examples based on the density distribution. This cluster ordering can be used by a broad range of density-based clustering, such as DBSCAN. And besides, OPTICS can provide density information about the dataset graphically by cluster reachability-plot [18], which makes it possible for the user to understand the density-based structure of dataset.

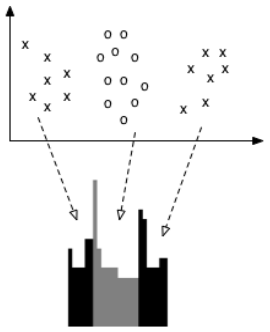


Fig.1 2-D dataset sample and corresponding reachability plot

Fig.1 gives the reachability-plot of the 2 dimensional dataset and the number of the valleys indicate that there are 3 density-based clusters.

However, OPTICS needs some priori, such as neighbourhood radius (ε) and a minimum number of objects (MinPts) within ε, by which directly density-reachable, density-connected, cluster and noise are defined as in [18].

DBSCAN [19] is based on density-connected range from arbitrary core objects, which contains MinPts objects in ε-neighbourhood. In OPTICS, cluster membership is not recorded from the start, but instead the order in which objects get clustered are stored. This information consists of two values: core-distance and reachability-distance. For more details on DBSCAN and OPTICS ordered dataset are provided in [18].

Core-distance of an object p is defined as:

$$core-dis\tan ce_{\varepsilon,MinPts}(p)=\begin{cases}Undefined, if\left|neighbour_{\varepsilon}(p)\right|<MinPts\\ MinPts-dis\tan ce(p), otherwise\end{cases}$$

Reachability-distance of an object q w.r.t object o is defined as:

$$reachability-dis\tan ce_{\varepsilon,MinPts}(q,o)=\begin{cases}Undefined, if\left|neighbor_{\varepsilon}(o)\right|<MinPts\\ \max(core-dis\tan ce(o), dis\tan ce(o,q)), otherwise\end{cases}$$

Since reachability plot is insensitive to the input parameters, [18] suggests that the values should be "large" enough to yield a good result with no undefined examples and reachability-plot looks not jagged. Experiments show that MinPts uses values between 10 and 20 always get good results with large enough ε. Briefly, reachability-plot is a very intuitive means to get the understanding of the density-based structure of financial data. Its general shape is independent of the parameters used.

C. Data stream clustering

[9] applied an on-line evolving approach for detecting of financial statements' anomalies. The on-line evolving method [20] is a dynamic technique for clustering data stream. This method dynamically increases the number of clusters by calculating the distance between examples and existing cluster centres. If this distance is higher than a threshold value, a new cluster is created and initialized by the example. This clustering algorithm can be summarised in three main steps:

(1) Calculate the distance D_{iJ} between data object x_i to all existing cluster centres Cc_J, find the minimum distance D_{ik} and compare it to the radius R_k of cluster C_k.

(2) If $D_{ik} < R_k$ then x_i belongs to cluster C_k, else find the nearest cluster C_a and evaluate $S_a = D_{ia} + R_a$ against a threshold δ.

(3) If $S_a > \delta$ then create a new cluster for x_i else x_i belongs to cluster C_a and update $R_a = S_a/2$.

In this algorithm, the number of clusters is not predefined. However, the distance calculation and the threshold value needs expert to provide prior knowledge and so does label of newly formed cluster.

[21] applied a hierarchical agglomerative clustering [2] approach to analyse stock market data. The authors proposed an efficient metric for measuring the similarity between clusters; a key issue for hierarchical agglomerative clustering methods. This similarity between two clusters $C = \{C_1, C_2, \ldots C_k\}$ and $C' = \{C'_1, C'_2 \ldots C'_k\}$ is defined as follows:

$$Sim(C,C')=(\sum_{i}\max_{j}Sim(C_i,C'_j))/k$$

where

$$Sim(C_i,C'_j)=2\frac{\left|C_i\cap C'_j\right|}{\left|C_i\right|+\left|C'_j\right|}$$

The authors also mentioned that some pre-processing techniques such as mapping, dimensionality reduction and normalisation should also be applied to improve the performance. Moreover, they used Precision-Recall method [21] to increase the cluster quality.

[7] also applied [21]'s approach for analysing financial data i.e. stock market. Besides, the authors defined a new distance metric based on the time period to cope with time series data. Concretely, the distance between stock i and stock j is given by:

$$d(i,j)=\left\|P_i-P_j\right\|_2$$

where

$$P_i(t)=\frac{s_i(t+1)-s_i(t)}{s_i(t)}\times 100$$

$s_i(t)$ is the stock value i at time t. The authors stated that hierarchical agglomerative clustering fed by normalised percentage change after filtering outliers gives the best result. However the identification of outliers needs a priori threshold. Moreover, the authors combine neural networks and association analysis with the clustering technique to analyse stock market datasets.

IV. EVALUATION AND ANALYSIS

A. Datasets

Different financial datasets have been discussed in this section. Some of the $R_{i,j} = \frac{S_i + S_j}{M_{i,j}}$ were selected by the authors' approaches. For instance, [17] used data obtained from Morningstar including 904 different funds classified in seven different investment objectives: World Wide Bonds, Growth, SMEs, Municipal NY, Municipal CA, Municipal State and Municipal National. Each fund has 28 financial variables and all are normalised before analysis. Meanwhile, [9] used synthesis datasets with 1000 documents containing financial statements. In [21] the authors used Standard and Poor 500 index historical stock dataset. There are 500 stocks with daily price and each stock is a sequence of some length *l* where $l \leq 252$. In [7] they analysed stock price datasets from 91 different stocks, which can be found at link http://finance.yahoo.com. The data covers three years; from November 1, 1999 to November 1, 2001.

We analyse moreover two financial datasets with k-means and density-based clustering approaches: German credit card and Churn. Both of these datasets are provided by UCI machine learning repository [22]. German credit dataset contains clients described by 7 numerical and 13 nominal attributes to good or bad credit risks. The data contains 1000 sample cases. The Churn dataset is artificial but are claimed to be similar to real-world measurement. It concerns telecommunications churn and contains 5 nominal attributes, 15 numerical attributes and 3333 examples. We analyse the dataset without the help of nominal attributes for several reasons, e.g. numerical attributes are taken internally within the commercial activities or business market while nominal attributes are stated by external concepts defined by market experts, whose significance is not promised. Moreover, nominal attributes are usually hierarchically dependent and can be missing while data mining models should have the capability to bypass these optional constraints to understand the structure of sample cases.

B. Criteria

The criteria used to evaluate clustering methods depend on each approach. For instance, [17] applied a relevant value of k by using the formula (1) and then discuss on results obtained from the running of k-means algorithm to classify mutual funds.

[7] uses normalised change $P_i(t)$ of stock i to overcome the discrete essence of time and difficulties to treat deviations or first difference of prices due to the wide range of possible stock prices. External clustering statics such as entropy and purity are used to define the closeness within an industry, and internal statistics such as separation and the silhouette coefficient to tell what degree the industries' are separate from each other.

[9] does not give a clustering criterion but claims that their work is the first step to building robust financial statements' anomaly detection system but it highly depends on the operator monitoring the process.

In this paper, we use well-known internal criteria to evaluate the clustering behaviour. Davies-Bouldin Index (DBI) [23] is used as a first internal criterion for clustering, which is defined as follows:

$$DBI = \frac{1}{N}\sum_{i=1}^{N} D_i$$

where N is the number of clusters and D_i is the tightness criteria of a cluster C_i, which takes the worst case scenario and is defined as:

$$D_i = \max_{j;i\neq j} R_{i,j}$$

where i and j are cluster indexes, $R_{i,j}$ is summary evaluation of two clusters of ratio between sum of tightness of two clusters and looseness between two centres.

S_k is the average internal Euclidean distance of the cluster indexed by k, and $M_{i,j}$ is the Euclidean distance between two clusters.

$$S_i = \sqrt[2]{\frac{1}{T_i}\sum_{j=1}^{T_i}\left|X_j - A_i\right|^2}$$

$$M_{i,j} = \left\|A_i - A_j\right\|_2$$

where A_i is the centroid of the cluster C_i, T_i is the size of C_i, X_i is an n dimensional feature vector assigned to C_i. The smaller DBI value is, the more efficient clustering is.

Dunn index (DI) is used as a second internal criteria for clustering, which is defined by:

$$DI = \min_{1\leq i\leq N}\left\{\min_{1\leq j\leq N, j\neq i}\left\{\frac{\delta(A_i, A_j)}{\max_{1\leq k\leq N}\Delta_k}\right\}\right\}$$

where Δ_k is various types of size notation of a cluster, it could be farthest two points in side a cluster, mean distance between all pairs or distance of all the points from the mean.

$$\Delta_k = \max_{x,y\in C_k}\left|x - y\right|$$

and $\delta(A_i, A_j)$ is the closest distance between clusters

$$\delta(A_i, A_j) = \min_{x_i\in C_i, x_j\in C_j, i\neq j}\left|x_i - x_j\right|$$

Unlike DBI, the larger DI is the better is the clustering. It evaluates the inter-cluster and intra-cluster distances. However, like DBI, the best clustering loses most general structural information about the dataset.

The main difference between DBI and DI is that DBI indicates the average tightness while DI is a worst-case

indicator.

C. Partitioning Methods

As in [17] group mutual funds with different investment objectives, they claimed that cluster analysis is able to explain non-linear structural relationships among unknown structural dataset. They found that over 40% of the mutual funds do not belong to their stated categories, and despite the very large number of categories stated; three groups are very important. Clustering helps simplifying the financial data classification problem based on their characteristics rather than on labels, such as nominal labels (customer gender, living area, income or the success of the last transaction, etc.). Besides, nominal labels may be missing or not provided. Thus our effort is to understand the detailed structure of financial data classification without the given class labels.

We give the DBI and DI of K-Means clustering of both normalised and un-normalised two datasets (German credit dataset and Churn dataset) to figure out what are the optimal k values for given datasets. To avoid information overfitting and loss of generality, we test k from 2 to 20. We normalise the attributes values between [0:1] in order to avoid large-scale attributes dominating the dataset features.

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

where the x_{max} and x_{min} are the max and min value of rescaled attribute.

From Fig.2, k=12 is optimal by DBI and k=8 is the optimal value by DI for the original German credit dataset, k=8 is the optimal value for the normalised German credit dataset by both DBI and DI. From the result, we know that attribute scale affects the clustering evaluation since the DI of clustering original dataset is around 0. Normalisation unifies the results of both average tightness and worst case.

From Fig. 3, k=12 is optimal by DBI and k=17 by DI for original churn dataset. k=2 is the optimal value by both DBI and DI for normalised dataset. Again, we notice that normalisation unifies the optimal clustering scheme while original attribute scale giving two clustering solutions.

Fig. 5 shows that normalised German credit dataset is well density distributed. When MinPts=10, by setting reachability-distance equal to 0.33, the dataset is partitioned into 23 density-based clusters and 1 noise cluster. There are 841 valid examples and 159 noise examples. When MinPts = 20, with the same reachability distance, dataset is partitioned into 15 density closed clusters and 1 noise cluster. There are 681 valid examples and 319 noise examples.

Despite the visualization of density distribution, from Table I, the clustering suffers from large proportion of noise and larger DBI values and lower DI values compared to K-means clustering. We can conclude that German credit dataset is more suitable for centroid-based clustering rather than density-based clustering.

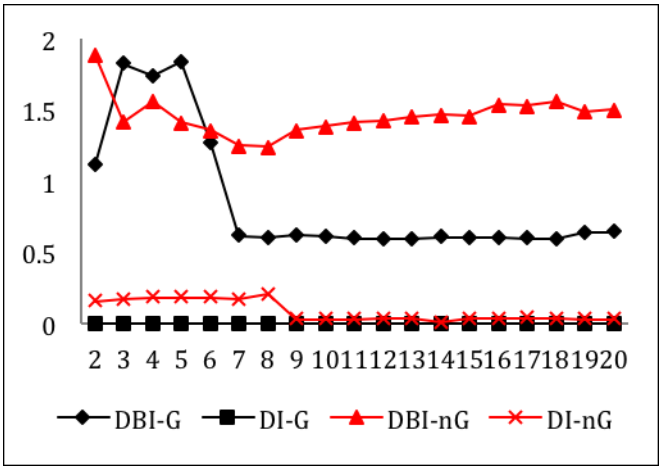


Fig.2 DBI and DI of K-means clustering German dataset

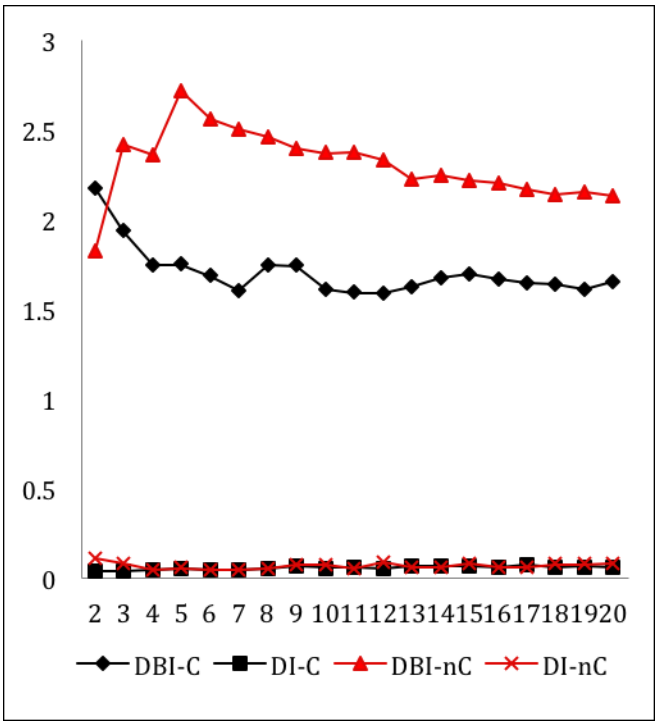


Fig.3 DBI and DI of K-means clustering churn dataset

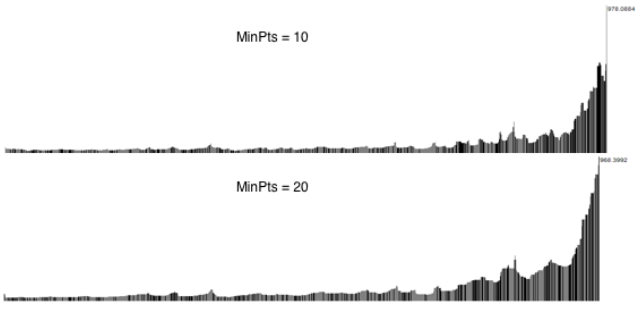


Fig. 4 Reachability-plot of original German credit dataset

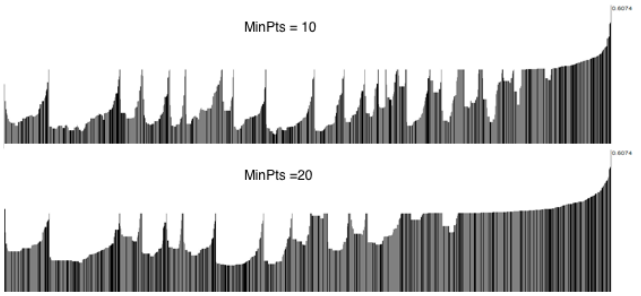


Fig. 5 Reachability-plot of normalized German dataset

Table I. DBSCAN clustering for normalized German credit dataset

Reachability distance	MinPts	Noise	DBI	DI
0.33	10	No	2.529	0.236
		Yes	2.843	0.033
0.33	20	No	2.465	0.250
		Yes	2.793	0.020

Fig.6 shows that original churn dataset cannot partitioned into clusters based on density; the entire dataset behaves as a whole.

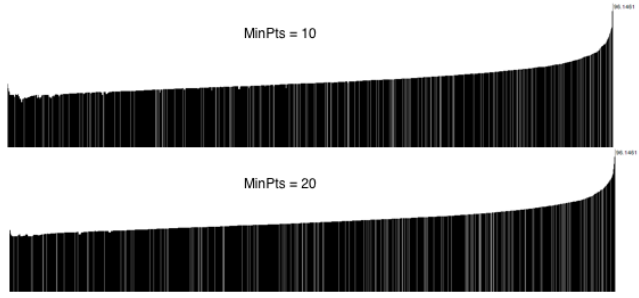


Fig. 6. Reachability-plot of original Churn dataset

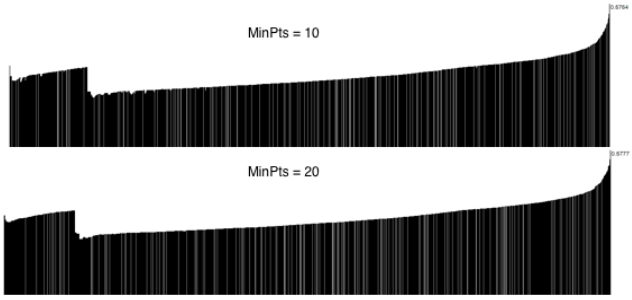


Fig. 7 Reachability-plot of normalized Churn dataset

Fig.7 shows that there are mainly two valleys when MinPts = 10 or 20, which indicates there are two incentive clusters in the churn dataset.

From Table II, DBSCAN without noise examples gets good DBI while getting poor DBI with noise. However, DBSCAN clustering suffers from large proportion of noise again, which has over 980 noise examples (around 30% of noise). For financial dataset, noise should be very small and the data recorded should be generally trusted. Financial datasets are not usually density distributed, and therefore, density-based clustering is not appropriate.

Table II. DBSCAN clustering and DBI for Churn dataset

Reachability distance	MinPts	Noise	DBI	DI
0.32	10	No	1.596	0.182
		Yes	3.568	0.106
0.33	20	No	1.572	0.195
		Yes	4.435	0.080

D. Data stream clustering

In [9] the authors use an on-line evolving clustering to update the parameters: cluster number and cluster radius. Two levels of anomalies detection have different financial statement features. The first level is based on internal information related to the account, e.g. equipment, employee, etc. For every combination of the two parameters, at least one cluster is created. But the authors do not give a good reason for it. The second level is based on document type. However the distances among different types are different, which is a prior knowledge from expert as well. The threshold values for creating new clusters are determined by the experts for the first level and pre-defined distance for the second level. The monitoring process involves experts heavily to approve or disapprove the documents as well. The authors categorise their method as the first step to anomalies detection. They are committed to reduce the reliance on experts and combine off-line and on-line approaches in the future work.

In [7] the authors use hierarchical agglomerative clustering for the time-based normalised stock market data. Percentage change is chosen to be a good comparative measure and time-based normalisation is used to remove the overall trend of stock market and improve the accuracy caused by outliers. The approach removes all items as outliers if the average normalised distance across all the items exceeds a specified threshold, which requires domain expert knowledge. Moreover, the degree of correlation of time-series is decided in advance. The authors found complete link and Ward's Method performs reasonably well by better purity and filtering out fewer outlier stocks. By treating the outlier, the overall purity decrease only about 6%, the author claims time-series clustering can determine the industry classification given the historical price record of a stock.

However, we notice that data stream clustering needs too much prior or domain knowledge and a lot of tuning for different features of even a single domain. Clustering approaches of different fields are different in essence. Thus clustering is a good method to understand the financial time-series classification but not logically clear and efficient. Distance measure becomes even more complex due to time related nature because clustering does not have the capability to scale time related influence intelligently between examples. Experts have to determine that instead, e.g. length of periodicity, etc. Recurrent neural networks [24] and Gaussian

Process [25] are more promising approaches and are more likely to handle time-series or periodical financial data classification.

V. CONCLUSION AND FUTURE WORK

We show that density-based clustering does not suit financial dataset. Normalised centroid-based clustering with higher DI or lower DBI gives the best number of clusters to help understanding financial data classification. Original attribute scales do not reflect the behaviour similarity since Euclidean distance is dominated by large scaled attributes, best average tightness does not indicate the best case by departing the worst case. However, we still find some constrains, e.g., K-means clustering tends to find spherical clusters, centroid-based clustering does not handle the noise, etc.

This work can be seen as the first step to look into the structure of financial dataset by using clustering. We would further apply other techniques on financial datasets. This includes: (1) discover other centroid-based clustering approaches for financial datasets. (2) Find if nominal attributes are significant and introduce other criteria to evaluate the clusters. (3) Introduce weighted Euclidean distance instead of standard Euclidean distance to re-evaluate centroid-based clusters, as to overcome the limitations of K-means. (4) Introduce and compare different kinds of nonlinear classifiers to strengthen the recall and accuracy and improve prediction, interpretability of the results. These techniques include decision tree, nonlinear SVMs, different structures of neural networks and Gaussian processes with different kernel functions, etc.

REFERENCES

[1] A. Weigend, "Data Mining in Finance: Report from the Post-NNCM-96 Workshop on Teaching Computer Intensive Methods for Financial Modeling and Data Analysis", *Fourth International Conference on Neural Networks in the Capital Markets NNCM-96*, 1997, pp. 399-411.

[2] P-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*, Addison Wesley, 2006, pp.150-172

[3] J. R. Quinlan, "Learning First-Order Definitions of Functions", *Journal of Artificial Intelligence Research.*, vol. 5, 1996, pp. 139–161

[4] N. Cristianini, J-S. Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods.* Cambridge University Press, 2000.

[5] J. Han and M. Kamber, *Data Mining: Concept and Techniques*. Morgan Kaufmann publishers, 2nd Eds., Nov. 2005.

[6] T. M. Cover, P. E. Hart, "Nearest Neighbor Pattern Classification", *Journal of Knowledge Based Systems*, vol. 8 no.6, 1995, pp. 373–389

[7] T. Wittman. (2002, December). Time-Series Clustering and Association Analysis of Financial Data. Available: http:// www.math.ucla.edu/~wittman/thesis/project.pdf.

[8] H. Bensmail, R. P. DeGennaro. (2004, September). Analyzing Imputed Financial Data: A New Approach to Cluster Analysis. Available: http:// www.frbatlanta.org/filelegacydocs/wp0420.pdf.

[9] S. Omanovic, Z. Avdagic, S. Konjicija, "On-line evolving clustering for financial statements' anomalies detection", *International Symposium on Information, Communication and Automation Technologies, ICAT 2009. XXII*, 2009, pp. 1-4.

[10] P. Langley, W. Iba, K. Thompson, "An analysis of Bayesian classifiers", *10th National Conference on Artificial Intelligence,* 1992, pp. 223-228.

[11] R. A. Bourne, S. Parsons, "Maximum Entropy and Variable Strength Defaults", *16th International Joint Conference on Artificial Intelligence, IJCAI 99*, Stockholm, Sweden, July 31 - August 6, 1992, pp.50-55

[12] N-A. Le-Khac, M. T. Kechadi "Application of Data Mining for Anti-money Laundering Detection: A Case Study". *10th IEEE International Conference on Data Mining Workshops*, Sydney, Australia, 14 December 2010. pp.577-584

[13] S. R. Eddy, "What is dynamic programming?", *Nature Biotechnology*, vol. 22, 2004, pp.909–910.

[14] R. S. Sutton, "Learning to predict by the method of temporal differences". *Machine Learning*, vol. 3, 1988. pp.9–44

[15] H. Wenying, "Wavelet Regression With an Emphassis on Singularity Detection," M.S. thesis, Dept. Mathematics and Statistics, Sam Houston State Univ., Texas, USA, 2003

[16] J.A. Hartigan, *"Clustering Algorithms",* Wiley 1975

[17] A. Marathe A, HA. Shawky ," Categorizing mutual funds using clusters", *Advances in Quantitative Analysis of Finance and Accounting*, vol. 7, 1999, pp.199–211.

[18] M. Ankerst, M. M. Breunig, H-P. Kriegel, J. Sander "OPTICS: Ordering Points To Identify the Clustering Structure*". ACM SIGMOD international conference on Management of data"*, 1999. pp. 49–60.

[19] M. Ester, H-P. Kriegel, J. Sander, X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise" 2nd *International Conference on Knowledge Discovery and Data Mining (KDD-96)*. 1996. pp. 226–231

[20] N. Kasabov, *"Evolving connectionist systems"*, Springer-Verlag London Berlin Heidelberg, 2003, pp. 40–42.

[21] M. Gavrilov, D. Anguelov, P. Indyk, and R. Motwani. "Mining the Stock Market: Which Measure is Best?" *Proc. of the KDD 2000*, p. 487-496.

[22] Professor Dr. Hans Hofmann, Statlog (German Credit Data) Data Set, C. L. Blake and C. J. Merz, Churn Data Set, UCI Repository of Machine Learning Databases

[23] Davies, D. L.; Bouldin, D. W. (1979). "A Cluster Separation Measure". *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2): 224.

[24] Martin T. Hagan, H. B. D., and Mark Beale. Neural network design. Boston, MA, USA, PWS Publishing Co. 1996

[25] MacKay, D. J. C. "Gaussian Processes - A Replacement for Supervised Neural Networks?". 1997

Consensus clustering from experts' partitions for patients' nevi: Model the Ugly Duckling

Y. Wazaefi[1], Y. Bruneu[2], J. Lefèvre[1], G. Menegaz[3], G. Paggetti[3], A. Le Troter[1], S. Paris[1], JJ. Grob[2], and B. Fertil[1]

[1] Laboratory of Sciences and Information Systems, Aix-Marseille University, Marseille, France
[2]Department of dermatology, La Timone Hospital, Marseille, France
[3]Department of Computing, University of Verona, Verona, Italy

Abstract - *Ugly duckling (UD) concept assumes that nevi in the same patient tend to share some morphological features so that dermatologists identify a few similarity clusters. UD is the nevus that does not fit into any of those clusters, likely to be suspicious. Our research program was to model the ability of dermatologists to identify the perceived similarity clusters (PSC) as an additional tool for computer-aided melanoma diagnosis systems. In the present study, nine dermatologists participated to do the clustering of nevi and to identify the UDs using dermoscopic images of nevi of 80 individuals. Dermatologists identified all confirmed melanomas as UDs and tend to be concordant about the identification of PSCs. We combined the multiple clusterings of dermatologists to find the consensus clustering, which yields a stable and robust final clustering. We demonstrated the limited variability of nevi patterns per individual (2.45 PSCs in average), whatever the number of nevi, which human brain has a natural intuitive ability to perceive.*

Keywords: consensus clustering, expert's agreement, melanoma diagnosis, Medicine Data Mining.

1 Introduction

Despite major progress in the treatment of advanced malignant melanoma (MM), prognosis of melanoma is still depending on our ability to detect these tumors as early as possible. There are several popular analytical methods to help the clinician to differentiate between benign and melanoma lesions, such as the ABCD rule [1], [2], the rule of "seven points of Glasgow Revisited" [3] for example. However, it has been argued that facing very similar pigmented skin lesions, dermatologists rather rely an unconscious process to detect MM, favoring in a first step intuition over analytical processes [4], [5] Whatever the approach, the diagnosis and subsequent management recommendations are based on morphological features, considered having absolute values, i.e. independent on the patient (morphological analysis).

The concept of Ugly Duckling (UD) introduced in 1998 [6] takes the patient context into account, implying that most nevi in a given individual tend to be similar, and can be classified into one or a few clusters of nevi sharing most morphological characteristics, the so called "Perceived Similarity Clusters" (PSC). It assumes that a lesion that does not fit into the main clusters of nevi of the given individual is suspicious. Conversely, a lesion that fits into the clusters of the individual is unlikely to be dangerous, even though it may be considered suspicious in terms of ABCD criteria. If the considered lesion shares its morphologic aspect with some of the other nevi of the individual, it should not be subjected to histological analysis. Grob and Bonerandi proposed that comparative analysis, illustrated by the concept of UD, is an important component in the ability to identify suspicious lesions, which has been confirmed by others [7], [8].

We designed an experimental study, to test the hypothesis that observers can build consistent PSCs in patients, in order to model this comparative process. Nine dermatologists participated to do the clustering of nevi and identify the UDs. Consensus clustering [9], also called aggregation of clustering, refers to the situation in which different clusterings have been obtained for a particular dataset from different inputs and it desired to find a single clustering which is better fit than the individual clusterings. Such problem was encountered for example in a knowledge reuse framework by Strehl and Ghosh (Market-baskets analysis) [10]. Consensus clustering is widely used in the domain of unsupervised learning (i.e. Clustering) to represent the consensus across multiple runs of a clustering algorithm (K-means, model-based Bayesian clustering, etc.), to determine the number of clusters in the data, to assess the stability of the clusters, and to reveal the significant differences between them.

In this part of study, a model of "consensus clustering" was built in order to make the best possible representation of nevi diversity in each patient. The ability of the brain to build unconsciously PSC, when the nevi of a patient are examined, is probably crucial for dermatologists' ability to detect UD and melanoma. Understanding this process may be very useful for the early diagnosis of melanoma, and for the development of computer-aided diagnosis systems.

2 Materials and methods

2.1 Data Collection

Digital dermoscopic images of all the pigmented skin lesions of 208 volunteers were collected between November 2009 and March 2011 (A total of 6249 images) from the French hospital "La Timone" at, Marseille. These 24-bit color RGB images in JPEG format were obtained with a numeric camera (SONY W120) attached to a dermoscope (HEINE Delta 20). An external dermatologist expert selected 80 individuals from the collected data with a total of 2089 images (mean: 26 images/individual; range: 8-81 images) to best represent the diversity of the entire dataset based on individual's age, sex, number and morphological diversity of pigmented skin lesions. This subset includes 7 histologically confirmed melanomas.

2.2 Observers

Nine observers, senior dermatologists specialized in nevi and melanoma, participated to this study. The panel of observers included five international leaders in the field from different countries, three of them leaders in the field of dermoscopy, a very experienced office-based dermatologist specialist of dermoscopy and three senior dermatologists from the research group.

2.3 Experiment

The experiment was conducted in a designated research room within the university of Aix-Marseille (École Supérieure d'Ingénieurs de Luminy) in Marseille, using iMac 21.5-inch (processor, 3.06 GHz; resolution, 1920 x 1080 pixels) and a secondary screen, Apple LED Cinema Display 24-inch (resolution, 1920 x 1200 pixels), for each observer. A web application has been created allowing a manual grouping of images. For each individual, a series of images was submitted directly to the screen so the observers can group these images by simple manipulations. Once the test has been completed for one individual, the observer was submitted a randomly new series of nevi of another individual.

The observers had to fulfill the experimental three-step process: The first step of the experiment (UD's identification step) explores the process of identification of UD. For each individual, the observer had to detect one or, more rarely, several UDs defined as "obviously different from individual's other nevi". The second step (clustering step) investigates the concept of PSC, in other words, the ability of human brain to recognize reliable morphological common characteristics among the nevi of an individual and to use them to identify the abnormal nevi. Observer was asked to group the nevi that are similar for a given individual, in as many clusters as needed.

The third step (hierarchical clustering step) studies the magnitude of similarity between nevi, as it is perceived by human brain. Observer was asked to merge the PSCs built at the second step, starting by the two most similar PSCs until getting a single cluster including all PSCs and UDs. The observer was asked to indicate the degree of reluctance to do each merging, as the difference perceived by him between the nevi of these clusters (low, medium, high and very high). This step provides a hierarchical clustering representing the degree of similarity between different nevi, as it is intuitively and subjectively assessed (The hierarchical clusterings obtained from this step weren't used in this paper).

The images were displayed with a low resolution such as 50 pixels per centimeter (clinical close-up), which refers to human visual perception of the morphological characteristics of the nevus unaided by a dermoscope, in order to obtain a clinical intuitive clustering experiment. Observers passed around 4 hours to complete the experiment.

2.4 Data analysis and statistics

2.4.1 Assessment of concordance between observers about UD

Cohen's Kappa (κ) is used here to measure the "concordance" between the observers, as far as UD is concerned. Cohen's kappa coefficient is a statistical measure of inter-rater agreement or inter-annotator agreement [11] for qualitative items. It is generally thought to be a more robust measure than simple percent agreement since κ takes into account the agreement occurring by chance. The similarities between the observers were represented in a similarity matrix of κ values. κ varies from 0, no agreement among the raters other than what would be expected by chance, to 1 where the agreement between the raters is complete. The interpretation of κ values is based in the scale published by Fleiss [12]: κ values of 0.00 to 0.40 represent poor agreement, 0.41 to 0.75 fair to good agreement, and a value above 0.75 is considered as almost excellent agreement.

2.4.2 Assessment of concordance between observers about PSC

This is a more complex issue, since observers' individual tendency to partition into more or less groups may jeopardize fundamental concordance between them. To compare the clustering between observers, we used the BCubed metric defined in [13] as algorithm. Briefly, BCubed metric combines B^3 Precision (the averaged precision of all items) and B^3 Recall (the averaged recall of all items) of the given partitions using the F1-score (1).

$$F(R,P)=2\times\left(\frac{B^3\text{Precision}\times B^3\text{Recall}}{B^3\text{Precision}+B^3\text{Recall}}\right) \qquad (1)$$

Item precision represents how many items in the same cluster of the concerned item in partition 1 belong to the same cluster of this item in partition 2 (specificity by item) (See Fig. 1). Symmetrically, Item recall represents how many items from the same cluster of the item in partition 2 appear in the same cluster of this item in partition 1. B^3 supports the constraints presented by Amigo et al. in [14], which are very relevant to our problem. It values cluster homogeneity (items belonging to the same cluster in partition 1 should be grouped in the same cluster in partition 2), cluster completeness (items belonging to the same cluster in partition 2 should be grouped in the same cluster in partition 1), rag bag criteria (introducing disorder into disordered cluster is less harmful than introducing disorder into a clean cluster), and cluster size rather than number of clusters (small error in a big cluster should be preferable to a large number of small errors in small clusters).

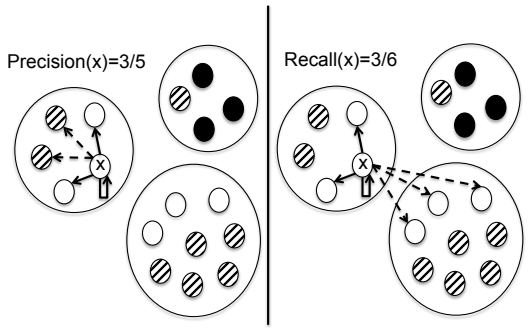


Fig. 1. Example of computing the B^3 precision and recall for one item (from Amigo et al. [12]). Circle shapes represent partition 1, and textures represent partition 2.

Other popular cluster validity metrics, such as, VI-measure [15], Adjusted rand index [16], Jaccard-index [17], and Mutual information [18], do not satisfy at least one of these constraints. B^3 permits to limit some potential biases in the assessment of concordance between partitions of different observers. These biases can be linked to the number and the size of initial clusters (clusters of the second step of this study), and the tendency to make a group with all nevi that cannot be clustered in a homogeneous group.

2.4.3 Extract the consensus clustering

The observers produced nine clusterings of the same individual. These clusterings were not identical, but had significant overlap. We need a "consensus clustering" that reconcile the nine observers partitions in order to represent the ability of observers to build consistent PSCs, and to model this process (i.e. the comparative analysis of nevi). Several approaches have been developed to solve the consensus clustering problems over recent years [18], [19–21]. To combine the multiple partitions of nevi of an individual into a single consolidated clustering, we need firstly to define a consensus matrix, which assess the similarity between the nevi. Following the Cluster-based Similarity Partitioning Algorithm (CSPA), proposed by Strehl and Ghosh [18], the similarity between a pair of nevi in an individual was estimated as the number of observer's partitions in which the two nevi were clustered together. In our hands, similarity varies from 0 to 9, since nine experts are concerned. For each individual, a matrix of similarity between all his nevi was calculated, and given as input to a hierarchical clustering algorithm [22], which seeks to create a hierarchy of clusters (dendrogram) grouping similar nevi. We evaluated the dendrogram obtained by four linkage criteria (single-linkage, complete-linkage, weighted average-linkage, and Ward's method [23–26]). In order to choose the appropriate method, we used the cophenetic correlation coefficient [27], which measures how faithfully a dendrogram preserves the pairwise distances between the original data.

In contrast to most clustering methods (i.e. K-means, K-medoids, etc.) there is no need to determine the number of clusters in the hierarchical clustering algorithm a priori. Nevertheless, the process of cluster detection in hierarchical clustering is referred to as tree cutting, or branch pruning. We should choose a cutoff through the dendrogram to represent the most natural division into clusters, leading to the "best" consensus clustering that can be derived from the nine observers' expertise. The most common unsupervised cutting method, the fixed height branch cut method, defines each contiguous branch below a fixed height cutoff as a separate cluster. Another way to cut the tree and extract the clusters, is based on the inconsistency coefficient method [28]. It associates a label to each link in the hierarchy by calculating the inconsistency coefficient of the link, which shows how much the two clusters connected by this link are similar. It compares the length of the link with the average length of other links at the same level of the hierarchy. The higher the value of this coefficient, the less similar the clusters connected by the link. [29], [30]. Cluster is preserved while the coefficient is lower than a predefined threshold. Clusters extracted in this way, do not necessarily correspond to a horizontal cut across the tree at a certain level, which is definitely more flexible. We adopt this unsupervised cutting method to find the clusters. We proposed also a supervised cutting of the hierarchy by applying the B^3 metric on each subtree of the hierarchy to find the optimal consensus clustering that had the highest agreement with the partitions of the nine observers.

2.4.4 Visualizing the similarity

Multidimensional scaling (MDS) [31] belongs to a set of related statistical techniques often used in information visualization for exploring similarities or dissimilarities in data. MDS is a special case of ordination. An MDS algorithm starts with a matrix of item–item similarities (or dissimilarities) and then assigns a location to each item in N-dimensional space, where N is *a priori* specified. This algorithm is used here to visualize the distances between the observers and between the nevi in two-dimensional and three-dimensional spaces.

3 Results

The number of nevi considered as UDs by observers is highly variable (Fig. 2), with a trend for some of them to see much more UDs than others. Observers see a mean number of one UD per individual (range, 0.49-2.04). There was an overall "fair to good" agreement between observers on the nevi identified as UDs, with a κ statistic of 0.50 in average (range, 0.43-0.55).

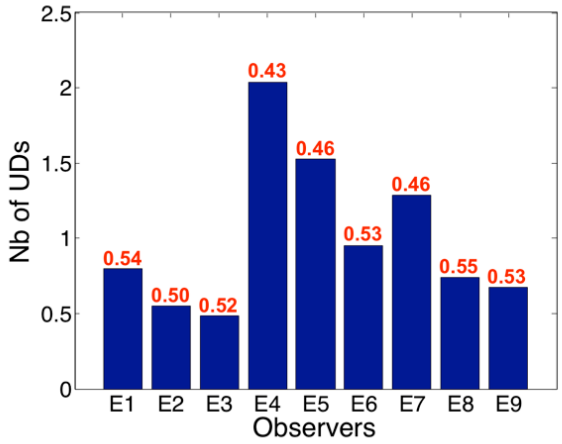


Fig. 2. Number of UDs identified by observers, with the agreement between each one and the others (mean pairwise κ; in red).

Of the 254 nevi identified as UDs by at least one observer, 54 (21% of total UDs) were identified as such by at least 5 observers, and 20 (8% of total UDs) by all 9 observers. In this study, nevus was considered as UD if it was perceived as such by at least 5 observers (consensual UD). All 7 melanomas were considered as UDs by the nine observers and subsequently as consensual UDs (sensitivity 100%), whereas only 47 of the 2082 benign nevi were perceived as consensual UDs (specificity 97.75%).

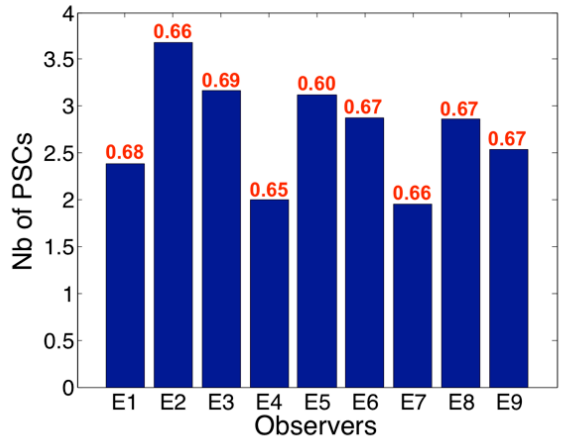


Fig. 3. Number of PSCs identified by observers, with the agreement between each one and the others (measured by B^3; in red).

The observers identified 2.73 PSCs in average per individual (Fig. 3). Differences between observers are smaller than for UDs. There was a good interobserver agreement on the created PSCs (B^3 mean, 0.66; range, 0.60-0.68). A statistically significant positive correlation is observed between the mean number of identified PSCs per individual and the number of nevi per individual but the net increase in the number of PSCs with the number of nevi per individual is moderate: 1.95 to 3.67 mean PSCs identified per individual face to 8 to 80 nevi per individual. The average values over all individuals of the cophenetic coefficient correlation for hierarchy was 0.81, 0.86, 0.89, 0.84 for the single-linkage clustering, the complete-linkage clustering, the weighted average-linkage, and the ward's method, respectively.

TABLE I
Unsupervised Cutting towards a Consensus Clustering

Methods	PSC	B^3	UD	Sensitivity UD	Sensitivity MM
Single	3.2	0.6085	3.5	50%	57.1%
Complete	6.1	0.5809	1.5	53.7%	71.4%
W-Average	6.3	0.5768	2	66.7%	85.7%
Ward	7	0.5575	0.8	31.5%	71.4%

PSC: Average number of PSCs identified per individual; B^3: Average agreement with the observers, measured by B^3; UD: Average number of identified UDs per individual. Sensitivity UD: Sensitivity for consensual UDs detection, Sensitivity MM: Sensitivity for MMs detection.

Table I shows the results of applying an unsupervised cutting on these hierarchies, setting the inconsistency coefficient threshold to one. The weighted average clustering had identified 6 melanomas out of 7 as different from other nevi (sensitivity 85.7%). The sensitivity of this method for consensual UDs detection was 66.7%. Single-linkage clustering was the method with the highest agreement with the clusterings of the nine observers (B^3, 0.61).

TABLE II
Supervised Cutting towards a Consensus Clustering

Methods	PSC	B^3	UD	Sensitivity UD	Sensitivity MM
Single	2.25	0.7583	2.8	100%	100%
Complete	2.64	0.7517	1	94.4%	100%
W-Average	2.45	0.7582	1.5	100%	100%
Ward	2.64	0.7485	0.8	85.2%	100%

PSC: Average number of PSCs identified per individual; B^3: Average agreement with the observers, measured by B^3; UD: Average number of identified UDs per individual. Sensitivity UD: Sensitivity for consensual UDs detection, Sensitivity MM: Sensitivity for MMs detection.

We applied the supervised cutting, using B^3 metric on the same hierarchies obtained by the four linkage methods (Table II). The results of weighted average-linkage clustering were the most convincing with 100% sensitivity for MMs detection as well as for consensual UDs detection. The four methods had a good agreement with the clusterings of the nine observers.

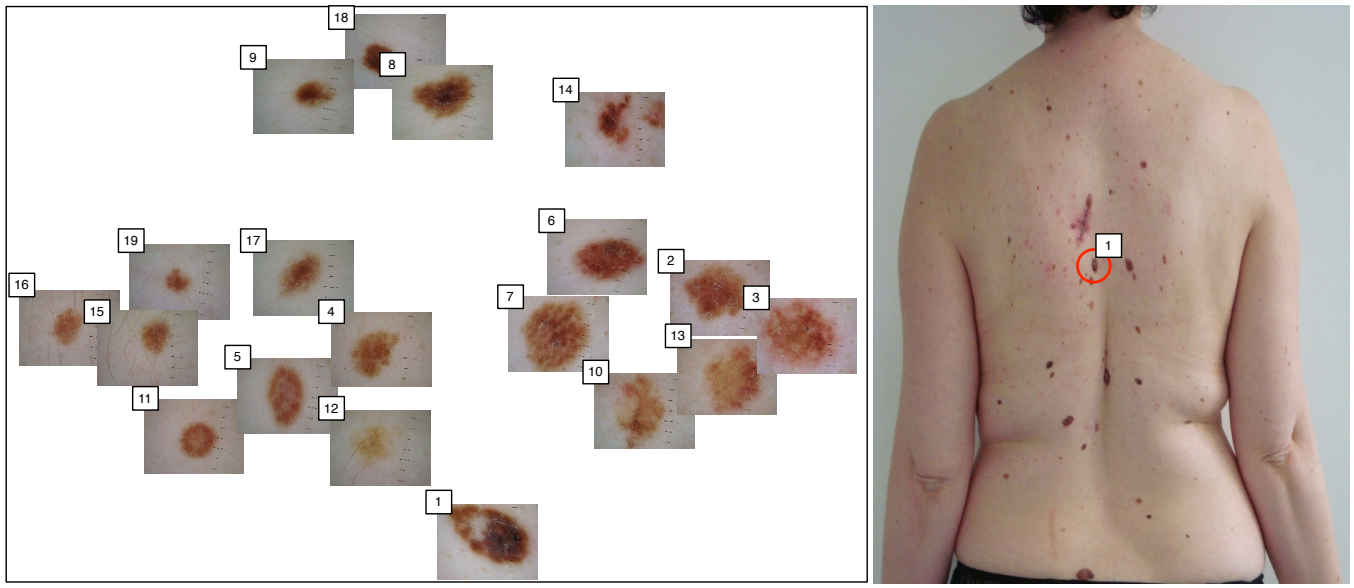


Fig. 4. Clinical close-up of the consensus clustering (left panel) of an individual with 19 nevi (right panel). The consensus clustering obtained by a supervised cutting on the weighted average-linkage hierarchical clustering and projected in two-dimensional space using MDS. The method had identified the two consensual UDs (lesion 1 and lesion 14) as different from patient's other nevi. Lesion 1 (right panel) was a malignant melanoma histologically confirmed that was apparent as UD for nine observers as well as for the consensus clustering (left panel).

Fig. 4 illustrates an example of a patient with 19 nevi (2 consensual UDs, 1 MM confirmed histologically) with the consensus clustering obtained by a supervised cutting applied on the hierarchy of weighted average-linkage clustering. The agreement between the consensus clustering and the clusterings of the nine observers was 0.75, and the cophenetic coefficient was 0.95. The consensus clustering identified the 2 consensual UDs as different from individual's other nevi.

4 Discussion and Conclusion

In this paper, we conducted an experimental study to test the ability of nine dermatologists to build consistent clusters in patients, according to the Ugly Duckling concept. We showed that, whatever the number of nevi in a patient, observers were able to reduce morphological diversity of nevi to a mean of 2.45 main patterns (PSCs) at clinical examination, with a good inter-observer concordance. Their sensitivity for the identification of malignant melanomas as different from patient's other nevi was 100%. The nine dermatologists have provided different partitions but no absolute gold standard (i.e. consensus clustering). When different input partitions differ significantly, the consensus by simple averaging is really a brut-force voting and there is no real "consensus" in their original meaning. We proposed a method to establish the consensus clustering of experts' partitions by using hierarchical clustering on the consensus similarity matrices between nevi.

We applied a supervised cutting on the hierarchy, using B^3 metric, to find the clustering with the highest agreement with the nine experts. We evaluated the hierarchical clusterings of four linkage criteria. The weighted average-linkage method was the most convincing with the highest agreement with the observers and the highest cophenetic correlation value. The sensitivity of this method for the melanoma detection was 100%. The consensus similarity matrices between nevi, in this paper, were calculated from the clusters created at the second step of the experiment. In future work, we wish to use the clusters created at the third step of the experiment (with the degree of similarity between clusters) to extend the consensus function for hard clustering used in this study (CSPA) to a consensus function for soft clustering.

The consensus clustering provides a model of the diversity of nevi in each patient, which can be used to find the different suspicious nevus (i.e. ugly duckling). Such information can be used to optimize melanoma diagnosis by comparative analysis and ugly duckling sign, and to enhance the computer-aided diagnosis. As already pointed out by Argenziano et al., a system that would make a decision from the analysis of all the nevi of a patient, and will be able to determine whether a nevus is different from patient's other nevi, is likely to be much more efficient and specific than a system based just on the analysis of single nevus features.

To model the comparative analysis of multiple nevi, we should learn the similarity measure relative to the consensus clustering. The problem of training a clustering algorithm to produce desirable clusterings is known as supervised clustering: given sets of items and complete clusterings over these sets, we learn how to cluster future sets of items [32], [33]. In our future work, we wish to use the consensus clustering extracted in this part of the study in a supervised clustering method.

5 Acknowledgment

This research was partially supported by the ANR (Agence Nationale pour la Recherche), VISOON, and the ADEREM (Association pour le Développement des Recherches Biologiques et Médicales – CHR Marseille). The authors also thank Dr. Caroline Gaudy-Marqueste, Dr. Sandrine Monestier, Dr. Luc Thomas, Dr. Marie-Françoise Avril, Dr. Raoul Triller, Dr. Giovanni Pellacani, and Dr. Joseph Malvehy for their participation to this study.

6 References

[1] J. T. McCarthy, "ABCDs of melanoma," *Cutis*, vol. 56, no. 6, p. 313, 1995.

[2] L. Thomas, P. Tranchand, F. Berard, T. Secchi, C. Colin, and G. Moulin, "Semiological value of ABCDE criteria in the diagnosis of cutaneous pigmented tumors," *Dermatology*, vol. 197, no. 1, pp. 11-17, 1998.

[3] M. F. Healsmith, J. F. Bourke, J. E. Osborne, and R. A. C. Grahambrown, "An Evaluation of the Revised 7-Point Checklist for the Early Diagnosis of Cutaneous Malignant-Melanoma," *British Journal of Dermatology*, vol. 130, no. 1, pp. 48-50, 1994.

[4] S. Girardi, C. Gaudy, J. Gouvernet, J. Teston, M. A. Richard, and J. J. Grob, "Superiority of a cognitive education with photographs over ABCD criteria in the education of the general population to the early detection of melanoma: a randomized study," *International journal of cancer*, vol. 118, no. 9, pp. 2276–2280, 2006.

[5] S. Girardi, J. Gouvernet, M. Richard, and J. Grob, "21 Randomized assessment of strategies for self-detection in the general population: failure of ABCD, success of cognitive approach," *Melanoma Research*, vol. 14, no. 4, p. A10, 2004.

[6] J. J. Grob and J. J. Bonerandi, "The 'ugly duckling' sign: Identification of the common characteristics at nevi in an individual as a basis for melanoma screening," *Archives of Dermatology*, vol. 134, no. 1, pp. 103-104, 1998.

[7] G. Argenziano et al., "Dermoscopy of Patients With Multiple Nevi: Improved Management Recommendations Using a Comparative Diagnostic Approach," *Archives of Dermatology*, vol. 147, no. 1, pp. 46-49, 2011.

[8] A. Scope et al., "The 'ugly duckling' sign: agreement between observers.," *Archives of dermatology*, vol. 144, no. 1, pp. 58-64, 2008.

[9] S. Monti, P. Tamayo, J. Mesirov, and T. Golub, "Consensus Clustering," *Machine Learning*, vol. 52, no. 1-2, pp. 1-34, 2003.

[10] A. Strehl and J. Ghosh, "A scalable approach to balanced, high-dimensional clustering of market-baskets," in *HiPC '00 Proceedings of the 7th International Conference on High Performance Computing*, 2000, pp. 525–536.

[11] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37-46, 1960.

[12] J. Fleiss, "Statistical methods for rates and proportions," *Wiley*, 1981.

[13] A. Bagga and B. Baldwin, "Entity-based cross-document coreferencing using the Vector Space Model," *Proceedings of the 36th annual meeting on Association for Computational Linguistics* -, vol. 1, p. 79, 1998.

[14] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo, "A comparison of extrinsic clustering evaluation metrics based on formal constraints," *Information Retrieval*, vol. 12, no. 4, pp. 461–486, 2009.

[15] M. Meila, "Comparing clusterings—an information based distance," *Journal of Multivariate Analysis*, vol. 98, no. 5, pp. 873-895, 2007.

[16] L. Hubert and P. Arabie, "Comparing partitions," *Journal of classification*, vol. 2, no. 1, pp. 193-218, 1985.

[17] P. Jaccard, "Etude comparative de la distribution florale dans une portion des Alpes et du Jura," *Bulletin del la Société Vaudoise des Sciences Naturelles*, vol. 37, pp. 547-579, 1901.

[18] A. Strehl and J. Ghosh, "Cluster Ensembles – A Knowledge Reuse Framework for Combining Multiple Paritions," *Journal of Machine Learning Research*, vol. 3, pp. 583-617, 2002.

[19] T. Li and C. Ding, "Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization," *Data Mining, 2007. ICDM 2007.*, vol. 2, no. 1, pp. 577–582, 2007.

[20] X. Z. Fern and C. E. Brodley, "Solving cluster ensemble problems by bipartite graph partitioning," in *Proceedings of the twenty-first international conference on Machine learning*, 2004, p. 36.

[21] A. Gionis, H. Mannila, and P. Tsaparas, "Clustering aggregation," *Journal ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, 2007.

[22] S. Johnson, "Hierarchical clustering schemes," *Psychometrika*, 1967.

[23] K. Florek, J. Kuraszewicz, J. Perkal, Steinhaush., and S. Zubryzki, "Sur la liaison et la division des points d'un ensemble fini," *Colloquium Mathematicum*, vol. 2, pp. 282-285, 1951.

[24] D. Defays, "An efficient algorithm for a complete link method," *The Computer Journal*, vol. 20, no. 4, pp. 364-366, 1977.

[25] J. Ward, "Hierarchical grouping to optimize an objective function," *Journal of the American statistical association*, vol. 58, pp. 236–244, 1963.

[26] W. H. E. Day and H. Edelsbrunner, "Efficient algorithms for agglomerative hierarchical clustering methods," *Journal of classification*, vol. 1, no. 1, pp. 7–24, 1984.

[27] R. R. Sokal and F. J. Rohlf, "The comparison of dendrograms by objective methods," *Taxon*, vol. 11, no. 2, pp. 33-40, 1962.

[28] C. Zahn, "Graph-theoretical methods for detecting and describing gestalt clusters," *Computers, IEEE Transactions on*, vol. 20, no. 1, pp. 86-86, 1971.

[29] M. Bailey, J. Oberheide, J. Andersen, Z. M. Mao, F. Jahanian, and J. Nazario, "Automated Classification and Analysis of Internet Malware," *Analysis*, pp. 178-197, 2007.

[30] M. Ghasemigol, H. S. Yazdi, and R. Monsefi, "A New Hierarchical Clustering Algorithm on Fuzzy Data (FHCA)," *International Journal*, vol. 2, no. 1, pp. 134-140, 2010.

[31] I. Borg and P. Groenen, "Modern multidimensional scaling: Theory and applications," *New York: Springer-Verlag*, pp. 207–212, 2005.

[32] T. Finley and T. Joachims, "Supervised clustering with support vector machines," in *Proceedings of the 22nd international conference on Machine learning*, 2005, pp. 217-224.

[33] T. Kamishima, S. Akaho, and F. Motoyoshi, "Learning from Cluster Examples-Employing Attributes of Clusters.," *Transactions of the Japanese Society for Artificial Intelligence*, vol. 18, no. 3, pp. 86-95, 2003.

Intrusion Detection System with Data Stream Clustering Approach

Madjid Khalilian[1,2], Norwati Mustapha[2], Md Nasir Sulaiman[2], Ali Mamat[2]

[1]Islamic Azad University, Karaj Branch, Iran
[2]Faculty of Computer science and information technology
University Putra Malaysia
khalilian@ieee.org, {norwati, nasir, ali}@fsktm.upm.edu.my

***Abstract*—*fast and high-quality Intrusion Detection algorithms play an important role in providing security management component by organizing large amounts of information into a small number of meaningful clusters. In particular, clustering algorithms that build meaningful groups of data via network log file are ideal tools for their interactive visualization and exploration as they provide a powerful mechanism to detect malicious sessions. This paper focuses on data stream algorithms that build such detection solution and (i) present a comprehensive study data stream clustering algorithm that use different functions and schemes to solve different problems in this area, and (ii) presents a new class of clustering algorithms called Divide and Conquer stream clustering algorithms, which combine features from both partitional and agglomerative approaches that allows them to reduce the early-stage errors made by agglomerative methods and hence improve the quality of clustering solutions. The experimental evaluation shows that, Proposed method lead to better solutions than previous algorithms; making it ideal for clustering large amount of datum network log file due to not only their relatively low computational requirements, but also higher clustering quality. Furthermore, the proposed method consistently leads to better solution when there is no cluster in a window of data and data is monotonous, as well.*

***Index Terms*—** Data mining, data stream clustering, Intrusion Detection, divide and conquer.

I. INTRODUCTION

During the last decade many applications have been developed that they should manage massive amount of data which causes limitation in data storage capacity and processing time. Furthermore, many applications must operate in real-time to achieve their objectives. As an important case for these kinds of application, Network Intrusion Detection System (NIDS) can be pointed where it generates a huge data and this data should be process in real time to discover suspicious data. However, some difficulties against this problem list as below:

- Stream data may only be visit once because of huge data and time constraint.
- Algorithm should be operated in resources constraints especially for memory.
- Data is monotonous during a period of time; consequently, data clustering is meaningless.
- Number of clusters may be unknown in advance and the characteristics of clusters may change over time.
- Novel attack detection.
- Many data stream mining methods such as classification need to construct a model to classify and detect malicious data which is obviously time consuming and not suitable for real time environments.

It is desirable to have algorithms which are able to detect clusters of objects with evolving intrinsic with considering this point that visiting of data is possible once. In addition, resource constraints and outliers detection are others aspects. Therefore, the main objective for this paper is to propose a method to overcome the above mentioned problems. Moreover, the parameters in extracting general components and determining the suitable quality measurements will be studied. This paper focuses on data stream clustering makes two key contributions.

First, motivated by recent advances in data stream clustering, we revisited the question of whether or not data stream clustering approaches generate superior clusters result than traditional or other approaches and performed a comprehensive experimental evaluation of proposed method using standard dataset. We compare two recently most popular studies that have been shown to produce high-quality solutions with the method that studied vector space model and silhouette criterion to employ for processing high scaled dataset. Our experiments show that proposed condensation-based method generates hierarchical clustering solutions that are consistently and substantially better than those produced by the previous algorithms. These results suggest that condensation clustering algorithms are ideal for obtaining effective solutions of large datasets due to not only their relatively low computational requirements, but also better performance in terms of cluster quality.

Second, we present a new class of condensation algorithm called Divide and Conquer stream clustering algorithm (DC-STREAM) in which we introduce micro clusters obtained by condensation clustering algorithm based on vector model to constraint the space over which agglomeration decisions are made. This algorithm generates the clustering solution by using divide-and-conquer method to build a hierarchical structure for each partition of cluster. Our experimental evaluation shows that these methods consistently lead to better solutions than recent methods. The rest of this paper is organized as follows. Section 2 provides an overview for most popular and recent studies for Intrusion

Detection with data mining approaches and data stream clustering methods. Section 3 explains some preliminary information on how divide and conquer are employed and how the vector space is represented. Section 4, 5 describes methodology as well as silhouette criterion and DC-STREAM algorithm. Section 6 provides the detailed experimental evaluation of the proposed data stream clustering method with taking into account limitations in Intrusion Detection dataset. Section 6 analyzes the impact of proposed method on the quality of the results. Finally, Section 7 provides some concluding remarks.

II. LITERATURE REVIEW

If we want to categorize intrusion detection methods, we will recognize two main aspects for grouping approaches, which one group refers to type of attack includes host based and network based. Another group of approaches refers to solutions techniques which are signature based and anomaly detection methods. In continue we review these techniques with their pros and cons.

Table 1 Intrusion Detection methods

Methods based on type of attacks	pros	cons
Host based[1]	HIDS monitor only the host, it can determined intrude more accurate. It does not need to install extra hardware or software because everything is on the host. Encrypted messages are not serious problem because they received in the host and can be decrypted more easily	It cannot detect some types of attacks that they need to monitor traffic of network e.g. DOS and DDOS. Redundancy is an important problem in HIDS especially when we want to install this system for a network, because we should have a HIDS for each host. Because of the fact that HIDS should be installed in each host, it is clear that expense of system will be increased.
Network based[2, 3]	Detection of some attacks such as DOS and DDOS need to monitor traffic of whole network and it is possible by NIDS. Low expense is brilliant advantage for NIDS because it is not necessary to install many monitoring systems.	Losing some data during the process of detection. Encrypted data are problematic in NIDS. In large-scale network more facility is required to monitor network; thus, scalability is another significant problem in NIDS.

On the other hand, there are two main solutions for IDS in terms of algorithms which are employed. We divide approaches in two main groups: misuse detection which the main study is the classification algorithms and anomaly detection which the main study is the pattern comparison (association rules and sequence rules) and the cluster algorithms.

Table 2 General approaches for IDS

approach	pros	cons
Signature based[4]	Specifying exact class of attacks. Efficiency is high and complexity is low.	Many false positives: prone to generating alerts when there is no problem in fact. Cannot detect unknown intrusions.
Anomaly[5, 6] detection	Anomaly detection can detect novel attacks to increase the detection rate. Compared to supervised methods, unsupervised approach breaks the dependency on attack-free training datasets.	Obviously, not all typical behaviors are attacks or intrusion attempts[7].

It is clear from table 2 that most efforts and solutions have been focused on anomaly detection methods. In other word, data mining methods are the most significant tools in this area. Among data mining approaches, data stream clustering has received most attractions due to its advantages in comparison of other methods. In fact, properties of data stream clustering such as being unsupervised, no need to have model, considering time and space constraints, working in high scaled datasets and eventually detecting novel class of data makes it most suitable to solve intrusion detection problem [8-13].

Generally, we can categorize data stream clustering from framework perspective in two main groups: on-line and off-line components; consequently, stream clustering take places in two steps on-line and offline. This kind of framework was proposed for the first time by [14] and in continue they have applied this framework to solve other problems in data stream clustering[15-19]. As it mentioned before main problems in data stream clustering are visiting data once and concept drift. Thus, Micro clustering and Macro clustering are utilized in two main components. It also employs a pyramid structure for organizing macro clusters during the time to answer user's question during tilted time and experimental results has demonstrated acceptable accuracy and efficiency. Generally speaking, approaches which are applied K-Means or K-Medians suffer from lack of accuracy when there are a lot of outliers. Beside, K-Means is also sensitive to value of outliers. These methods are not suitable for discovering clusters with non-convex shapes or clusters of very different size. Number of clusters should be determined as value of parameter K. Aforementioned weaknesses motivated researchers to employ some other techniques, e.g. [20] have developed a connectivity based reprehensive points to cluster data stream. Accuracy is outstanding in their research but it exhibits low performance. Another point is using a repository for previous data so it is unable to give us a history in different scale time. Ref[21] proposed a new framework for online monitoring clusters over multiple evolving streams by correlations and events. The streams are smoothed by piecewise linear approximation, and each end point of the line segment can be regarded as a trigger point. At each trigger point, for clusters that have trigger streams, they update the weighted correlations related to trigger streams in clusters. Whenever an event happens, the clusters are modified through efficient split and merge processes. In [22] a

new entropy based method has been developed for mixed numeric and categorical data stream clustering. They also use online and off line components to process data.

BIRCH can be considered as a primitive condensational method which is not component-based [23]. It works based on two steps: first it scans dataset and builds a tree which includes information about data clusters. In second step BIRCH refines tree by removing sparse nodes as outliers and concrete original clusters. STREAM is the next main method which has been designed especially for data stream clustering [24]. In this method K-Medians is leveraged to cluster objects with SSQ criterion for error measuring. There are some other approaches that they are not component-based methods[25-28]. We summarize the problems and solutions which are depicted in table 3.

Table 3 Major problems and solutions for data stream clustering

Problem	solution	pros	cons
Scan data once (time and space constraints)	Condensation-based[15 , 29-31]	Having summary of data (global view)	Resource constraints, speed up
	Data sampling[32]	speed up	Low quality
	Density -based[33]	Arbitrary shaped clusters	Applicable in low dimension
	Grid-based[34, 35]	Arbitrary shaped clusters	Applicable in low dimension
Evolving data and concept drift	Fading function(decay concept)[30, 34, 36]	Managing evolving data efficiently	threshold boundary value ,missing clusters
	Tree structure[37, 38]	No need to determining extra parameters	Inflexibility

In addition we have some minor problems in data stream clustering which are open issues in this area:

- ✓ High dimensional data
- ✓ Detecting noise and outliers
- ✓ Space constraints
- ✓ Uncertainty data
- ✓ Different data types
- ✓ Spherical shaped clusters vs. arbitrarily shaped clusters
- ✓ Number of determined parameters

III. PRELIMINARIES

Clustering is grouping samples into some classes unsupervised with unknown label. Let consider space with samples which are defined with vectors: $S = \{V_1, V_2, ..., V_n\}$ as each vector has own properties that is shown with v_1, $v_{2,...}$ so $V_{ij} \in R^d$ means j^{th} property of V_i and R^d is a space with d dimensions.

A. Divide AND Conquer method

When the size of a data set is too large to be stored in the main memory, it is possible to divide the data into different subsets that can fit the main memory and to use the selected cluster algorithm separately to these subsets. The final clustering result is obtained by merging the previously formed clusters. This approach is known as divide and conquer [39, 40]. On the other hand, most of the clustering techniques based on divide and conquer method ignore the fact about the different size or levels – where in most cases, clustering is more concern with grouping similar objects or samples together ignoring the fact that even though they are similar, they might be of different levels. For really large data sets, data reduction should be performed prior to applying the data-mining techniques which is usually performing dimension reduction. The main question is whether some of these kind of prepared and preprocessed data techniques can be discarded without sacrificing the quality of results. Due to this reason, each stream of data can be divided to some subsets based on their levels and clustering is applied on each subset instead of the whole stream of data.

B. Equivalency AND Similarity

A binary relation R on set S is called equivalence relation if and only if it is reflexive, symmetric and transitive.

A binary relation R on set S is called compatible relation if and only if it is reflexive and symmetric.

Claim: We can get a result from above definitions that each equivalence relation is also a compatible relation but a compatible relation is not necessarily an equivalence relation.

Let R be an equivalence relation on a set S. the equivalence class of $x \in S$ is the set $[x] = \{y \in S \mid yRx\}$, and the set of all equivalence classes of R is denoted by R(S).

Lemma.1. Each equivalence relation R can divide S into some partitions (classes) $s_1, s_2, \ldots, s_n$ where

$$a) \bigcup_{i=1}^{n} s_i = S$$

$$b) s_i \neq \varnothing$$

$$c) s_i \cap s_j = \varnothing, i \neq j$$

Proof can be seen in Discrete Mathematics Structure [43].

Lemma.2. Length of vector is an equivalence relation where length is defined by $L(V) = \sqrt{\sum_{i=1}^{d} v_i^2}$, where d and v_i are for number of dimensions and value of i^{th} feature respectively.

Proof:

reflexive

$$\forall V_i \in S : L(V_i) = L(V_i)$$

symmetric

$$\forall V_i, V_j \in S : L(V_i) = L(V_j) \Rightarrow L(V_j) = L(V_{ji})$$

transitive

$$\forall V_i, V_j, V_k \in S : L(V_i) = L(V_j) \wedge L(V_j) = L(V_k) \Rightarrow L(V_i) = L(V_k)$$

Outcome of lemma 1, 2 is the important result that implies length of vector can divide our problem space into some subsets with equivalency property. Although, vectors in one subset are in the same level but they might be in different directions. The radius of subset is L (V_i) for all vectors inside of that subset as V_i belongs to subset.

Lemma.3. Similarity is a compatible relation where similarity is defined by Minkowski distance[41]:

$$D_n(V_i, V_j) = \left(\sum_{k=1}^{d} \left| V_{ik} - V_{j_k} \right|^n \right)^{1/n}$$

Or cosine measure that is used inner product for similarity among vectors: $\cos(V_i, V_j) = \frac{V_i^T \cdot V_j}{|V_i||V_j|}$

Proof

$a)$*reflexive* $\therefore \forall V_i \in S : D_n(V_i, V_i) = 0, \cos(V_i, V_i) = 1$

$b)$*symmetric* $\therefore \forall V_i, V_j \in S : D_n(V_i, V_j) = D_n(V_j, V_i), \cos(V_i, V_j) = \cos(V_j, V_i)$

Similarity is not an equivalence relation because it doesn't have transitive property.

IV. METHODOLOGY

We are going to design an algorithm in two steps. First it divides entire space into some subsets based on length of vectors. As it mentioned before, samples in each subset are same size but not necessarily similar. For this purpose we cluster samples based on their size or length. In second phase K-Means algorithm is employed in each subset. It is suitable to overcome clustering problem for large datasets with high dimensions instead of using clustering on entire data.

Data Dividing: We divide arrival data in stream into some subsets by K-Means algorithm based on length of vector which is equivalency relation. Length of vectors are input for K-Means algorithm and output will be some partitions which elements inside them are same size and ready to clustering. In other word all samples in one partition have almost same size but might be dissimilar.

Subsets clustering: after finding subsets, clustering algorithm is applied on each subset and outcomes final group. Although samples in different subsets may be similar based on COSINE criterion but they are in different levels.

Silhouette refers to a method of interpretation and validation of clusters of data[41]. The technique provides a succinct graphical representation of how well each object lies within its cluster. Assume the data have been clustered via any technique, such as k-means, into clusters. For each datum *i*, let *a(i)* be the average dissimilarity of i with all other data within the same cluster. We can interpret *a(i)* as how good was *i* match to the cluster that it was assigned to (the smaller the value, the better the matching). Then find the average dissimilarity of *i* with the data of another single cluster. Denote the lowest average dissimilarity to *i* of any such cluster by *b(i)* . The cluster with this average dissimilarity is said to be the "neighboring cluster" of *i* as it is beside to the cluster *i* is assigned that is the cluster in which *i* fits best. We now define s(i), value of the silhouette for object i as below:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

In other word:

$$s(i) = \begin{cases} 1 - a(i)/b(i), & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{if } a(i) > b(i) \end{cases}$$

where $-1 \le s(i) \le 1$

Based on silhouette mean value we are able to compare quality of clusters from point of view both compactness and separateness. In addition, we utilize mean silhouette value criterion for finding the best value for number of clusters as it will be described further.

Dataset description: The 1998 DARPA Intrusion Detection Evaluation Program was prepared and managed by MIT Lincoln Labs. The objective was to survey and evaluate research in intrusion detection. A standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment, was provided. The 1999 KDD intrusion detection contest uses a version of this dataset. Lincoln Labs set up an environment to acquire nine weeks of raw TCP dump data for a local-area network (LAN) simulating a typical U.S. Air Force LAN. They operated the LAN as if it were a true Air Force environment, but peppered it with multiple attacks. The raw training data was about four gigabytes of compressed binary TCP dump data from seven weeks of network traffic. This was processed into about five million connection records. Similarly, the two weeks of test data yielded around two million connection records. A connection is a sequence of TCP packets starting and ending at some well defined times, between which data flows to and from a source IP address to a target IP address under some well defined protocol. Each connection is labeled as either normal, or as an attack, with exactly one specific attack type. Each connection record consists of about 100 bytes. Attacks fall into four main categories: 1) DOS: denial-of-service, e.g. syn flood; 2) R2L: unauthorized access from a remote machine, e.g. guessing password; 3) U2R: unauthorized access to local super user (root) privileges, e.g., various ``buffer overflow'' attacks; 4) probing: surveillance and other probing, e.g., port scanning.

Most of the connections in this dataset are normal, but occasionally there could be a burst of attacks at certain times. Also, each connection record in this dataset contains 42 attributes; whereas, all 34 continuous attributes will be used.

V. DIVIDE AND CONQUER STREAM CLUSTERING (DC-STREAM)

Our proposed algorithm for on-line component includes two main steps:

Step 1:

1. Compute length of vector for all samples in each group of data.
2. for I=2 to k find clusters with max value for average of silhouette, If this value is less than 0.25 then ignore subset dividing and return.
3. Return I as the number of subsets.

*Step2***:** In this step we can assume two different strategies for arriving and processing data with different results. First, data arrive one by one and assign it to suitable micro cluster. Therefore, managing evolving data and concept drift should be carried out by employing fading function and decay concept. Second, processing data can be done in batch; namely, Clustering must be done for entire data in one window. Thus, novelty detection is carried out automatically by finding the new micro clusters and concept drift is managed by frequent micro clusters statistics merging and splitting. In this study we employ second strategy to overcome data stream difficulties.

1. Input I as a number of subsets
2. If average of silhouette is more than 0.999 or I=0 then no need to cluster data in window and return entire of data in window as the one cluster. Go to next window of data.
3. for each subset s_i :

   Find the best number of clusters in each subset in the window based on value of the mean silhouette.

If mean of silhouette value is less than 0.25 for one cluster inside the subset and the total number of clusters<=k/I then re-cluster of this sub cluster.

Summarize and gather statistics information for each cluster as the micro cluster and add it to list of micro clusters.

4. Update list of micro clusters:
   - If number of samples in one micro cluster is less than threshold (minimum number of samples) which is determine by user then it is identifying as the outlier.
   - Merge micro clusters which not only their means but also mean of silhouette is near to each other.
   - Split micro cluster when its standard deviation becomes larger than minimum standard deviation or mean of silhouette is less than 0.25 into two micro clusters and then newly generated micro clusters are inserted into micro clusters set.
   - Identify micro clusters which have not received data during long time as the expired micro clusters.

VI. EVALUATION

The proposed method will be evaluated by number of subsets analysis and scalability.

Number of subsets analysis: one of the most important parameters which may significantly impact the clustering quality and speed up is the number of subsets in each window. As discussed earlier, this was defined based on length of vectors relationship and dividing data in window into some subsets of vectors which are in the same level. Finding best value for number of subsets can affect on time complexity and thus increase efficiency. We find this value by applying k-means algorithm five times for avoiding local minima in each window for length of vectors for 100000 samples and calculating mean of silhouette for different value of k from 2 to 10. As demonstrated in Figure 1, there is the most stable value for silhouette mean in k=2. Furthermore, null value or value close to 1 (0.999) for mean(s) imply that the entire data in window are same level and window of data can be considered as one cluster. Therefore, the number of subsets in each window was set to 2 for all experiments in this paper.

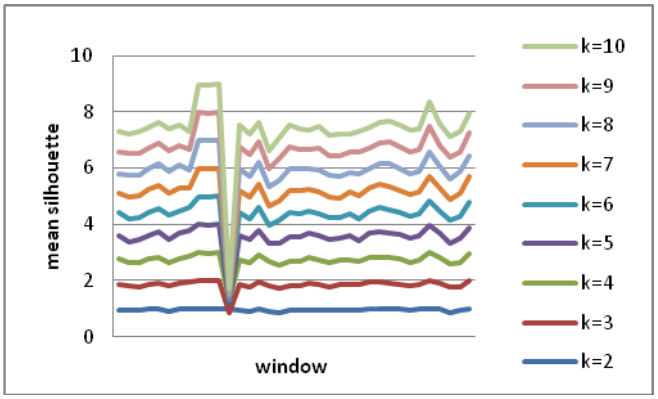


Figure 1 the mean of silhouette value for number of subsets in each window

We compare window 12, 13 as an instance to show how the number of subsets can help to group samples more efficient (Figure 2). All samples in window 12 are same size and it does not require to cluster but for window 13 only k=2 lead us to correct clustering.

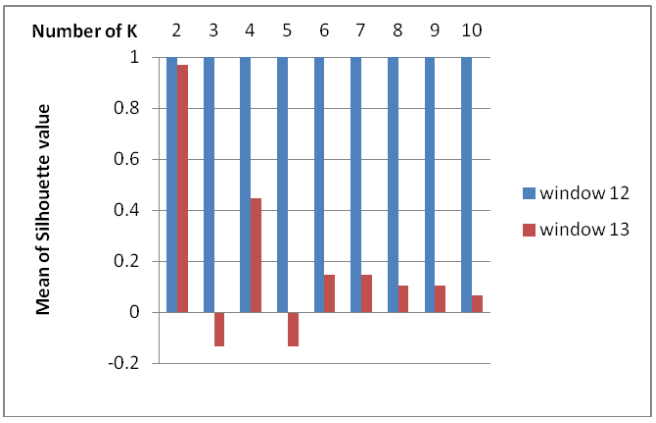


Figure 2 Mean of silhouette for window 12 and window 13 in difference values of k

One novel feature of DC-STREAM is it can create a set of micro clusters for each data window with considering both novelty and outliers. Furthermore, we expect this method to be more effective than current algorithms at clustering rapidly evolving data streams. We will show the effectiveness and high quality of method in detecting network intrusions. We compare the clustering quality of our method with STREAM and ConStream using the Network Intrusion dataset. All experiments for this dataset have shown that proposed method has substantially higher quality than STREAM and ConStream. Figure 3 shows some of our results, where stream speed = 1000 which means that the stream window length is 1000. We run each algorithm 5 times and compute their average of silhouette. As shown in the figure, DC-STREAM is always better than others. For example, at first window, the average silhouette of DC-STREAM is close to 1 whereas others achieved only 0.3 for mean(s). Surprisingly, the high clustering quality of DC-STREAM is achieved from its good design. On the one hand, the divide and conquer enables DC-STREAM to approximate a hierarchical structure based on level of objects as closely as desired. This is contrast with other clustering algorithms that only based on the k-means data stream clustering with its weaknesses such as initial value for clusters and outlier detection.

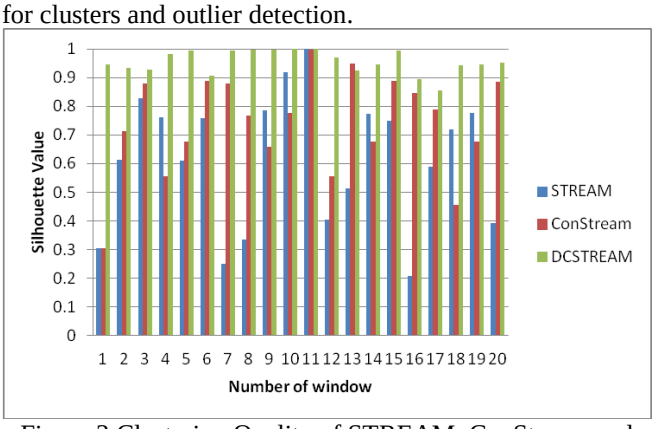


Figure 3 Clustering Quality of STREAM, ConStream and DC-STREAM for h=1000

Furthermore, an efficient method is required in order to maintain evolving data and concept drift. Therefore, our experiments also demonstrated that DC-STREAM is more

reliable than STREAM and ConStream whereby it always returns the same results in most of the time.

For example, at window 9, all the connections belong to the Smurf attack type. The micro-cluster maintenance algorithm always absorbs all data points in the same micro-cluster. As a result, DC-STREAM will successfully cluster all these points into one macro-cluster. This means that it can detect a distinct cluster corresponding to the network attack correctly.

Scalability results: The key to the success of the clustering method is high scalability of the micro-clustering algorithm. This is because this process is exposed to a potentially large volume of incoming data and needs to be implemented in an efficient and online fashion. On the other hand, some part of micro-clustering process required only a (relatively) negligible amount of time. This is because divide and conquers method in each window can reduce the size of problem and complexity. The most time-consuming and frequent operation during micro-cluster maintenance is on finding the micro clusters in batch. It is clear that the complexity of this operation increases linearly with the number of micro-clusters which is calculated based on mean(s) in order to obtain a high quality clustering.

VII. DISCUSSION

Our experiments have shown that DC-STREAM can facilitate cluster evolution analysis. By taking the Network Intrusion dataset as an example to show how such the analysis is performed in our experiments, we assume that the network connection speed is 1000 connections per each window. First, by comparing the data distribution for window start 9000 to 10000, all data is grouped into two clusters based on their length with mean(s) close to 0.999 so we can obtain only one micro clusters which includes all samples in the window. By checking the original dataset, we find that all samples in this kind of window (window 9) are attack connections (Smurf). More interestingly, all samples in window 9 are same size and type. Consequently, this step identifies the window as one micro cluster and the process is terminated. For other windows, there would be have the same situation and it will decrease time complexity. Another important point that should be mentioned here is differences between batch and online update. Although, batch processing in each window might be time consuming but it avoids some disadvantages such as determining extra parameters e.g. radius of micro clusters for detecting outliers or specific time for novel detecting. Split and merge which occur automatically after a window frame in the proposed method.

VIII. CONCLUSIONS

In this paper we have demonstrated some difficulties in data stream clustering which is its data mostly in high scale and high dimensions. New method need to be developed for processing these huge data sources. Furthermore, concept drift is nature of data and should also be managed by the new method. On the other hand, efficiency in terms of accuracy is one of the most critical measurements which are mostly defined by compactness and separateness for those data that their labels are unknown (for known labels we can use precision and recall). We discussed a new method for clustering and outlier detection of high scaled data with stream processing strategy. In order to achieve this goal, we used a compact representation of the clusters and a vector space model with considering divide and conquer approach which was utilized to construct an additive data stream mining algorithm. The resulting algorithm was applicable over Intrusion detection dataset with minor modifications. In addition, the technique can be applied to study the nature of the outliers and the evolution in the underlying stream. In addition, since the approach stores summary data about the clusters, it can be used in conjunction with a second level of clustering based on user-specified parameters. In most real applications, a stream may be monotonous during a specific window, thus entire of window can be considered as a micro cluster without any extra processing. For such cases, the first step of DC-STREAM can predict this situation and avoid further clustering which is particularly useful at the cost of processing time. The algorithm was tested on intrusion detection data sets. We found the algorithm to be highly effective in being able to quickly adapt to monotonous data in the data stream and recognize the number of subsets in each window. We also tested the method against the recent stream clustering methods known as CONStream and STREAM using silhouette measure that includes both compactness and separateness. In such cases, our method turns out to be much more effective, and the advantage was greater when the entire samples in the window are same size. As the future work, we will study global factor of the dataset and the relationship between successive subsets. On the other hand more experiments will be conducted in order to evaluating speed up and memory consumption.

REFERENCES

[1] Y. Yasami and S. Mozaffari, "A novel unsupervised classification approach for network anomaly detection by k-Means clustering and ID3 decision tree learning methods," *The Journal of Supercomputing*, vol. 53, pp. 231-245, 2010.

[2] D. Tian, Y. Liu, and Y. Xiang, "Large-scale network intrusion detection based on distributed learning algorithm," *International Journal of Information Security*, vol. 8, pp. 25-35, 2009.

[3] V. Sainani and M. L. Shyu, "A hybrid layered multiagent architecture with low cost and low response time communication protocol for network intrusion detection systems," 2009.

[4] D. Apiletti, E. Baralis, T. Cerquitelli, and V. D'Elia, "Characterizing network traffic by means of the NetMine framework," *Computer Networks*, vol. 53, pp. 774-789, 2009.

[5] A. Lazarevic, L. Ertoz, V. Kumar, A. Ozgur, and J. Srivastava, "A comparative study of anomaly detection schemes in network intrusion detection," 2003, pp. 25–36.

[6] M. Khalilian, N. Mustapha, M. N. Sulaiman, and A. Mamat, "Intrusion Detection System with Data Mining Approach: A Review," *Global Journal of Computer Science and Technology*, vol. 11, 2011.

[7] G. Singh, F. Masseglia, C. Fiot, A. Marascu, P. Poncelet, F. Guillet, G. Ritschard, D. Zighed, and H. Briand, "Mining Common Outliers for Intrusion Detection Advances in Knowledge Discovery and Management." vol. 292: Springer Berlin / Heidelberg, 2010, pp. 217-234.

[8] C. C. Aggarwal, "Data Streams: An Overview and Scientific Applications," *Scientific Data Mining and Knowledge Discovery*, pp. 377-397, 2009.

[9] A. Zhou, F. Cao, W. Qian, and C. Jin, "Tracking clusters in evolving data streams over sliding windows," *Knowledge and Information Systems*, vol. 15, pp. 181-214, 2008.

[10] A. S. H. Ismail, A. H. Abdullah, K. A. Bak, M. A. Ngadi, D. Dahlan, and W. Chimphlee, "A novel method for unsupervised anomaly detection using unlabelled data," 2008, pp. 252-260.

[11] S. Y. Jiang, X. Song, H. Wang, J. J. Han, and Q. H. Li, "A clustering-based method for unsupervised intrusion detections," *Pattern Recognition Letters*, vol. 27, pp. 802-810, 2006.

[12] W. Lee, S. J. Stolfo, P. K. Chan, E. Eskin, W. Fan, M. Miller, S. Hershkop, and J. Zhang, "Real time data mining-based intrusion detection," 2001, pp. 89–100.

[13] T. Subbulakshmi, G. Mathew, and S. M. Shalinie, "Real Time Classification and Clustering Of IDS Alerts Using Machine Learning Algorithms," *International journal of Artificial & Application*, vol. 1, p. 20.

[14] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A framework for clustering evolving data streams," 2003, pp. 81-92.

[15] C. C. Aggarwal and P. S. Yu, "A framework for clustering massive text and categorical data streams," 2006, p. 479.

[16] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A framework for projected clustering of high dimensional data streams," 2004, p. 863.

[17] C. C. Aggarwal and P. S. Yu, "A framework for clustering uncertain data streams," 2008.

[18] C. Aggarwal, "A framework for clustering massive-domain data streams," 2009, pp. 102-113.

[19] C. C. Aggarwal, "On high dimensional projected clustering of uncertain data streams," 2009, pp. 1152-1154.

[20] S. Lühr and M. Lazarescu, "Incremental clustering of dynamic data streams using connectivity based representative points," *Data & Knowledge Engineering*, vol. 68, pp. 1-27, 2009.

[21] M. Y. Yeh, B. R. Dai, and M. S. Chen, "Clustering over multiple evolving streams by events and correlations," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1349-1362, 2007.

[22] S. Wang, Y. Fan, C. Zhang, H. X. Xu, X. Hao, and Y. Hu, "Entropy Based Clustering of Data Streams with Mixed Numeric and Categorical Values," 2008, pp. 140-145.

[23] Zhang, Ramakrishnan, and L. M., "BIRCH: An efficient data clustering method for very large databases " in *ACM SIGMOD Conference on Management of Data*, 1996, pp. 103-114.

[24] L. O Callaghan, N. Mishra, A. Meyerson, S. Guha, and R. Motwani, "Streaming-data algorithms for high-quality clustering," 2002, pp. 685-696.

[25] I. Kiselev and R. Alhajj, "An adaptive multi-agent system for continuous learning of streaming data," 2008, pp. 148-153.

[26] G. Cormode, S. Muthukrishnan, and W. Zhuang, "Conquering the divide: Continuous clustering of distributed data streams," 2007, pp. 1036-1045.

[27] W. Jiang and P. Brice, "Data stream clustering and modeling using context-trees," 2009.

[28] A. Zhou, F. Cao, Y. Yan, C. Sha, and X. He, "Distributed data stream clustering: A fast EM-based approach," 2007, pp. 736-745.

[29] C. C. Aggarwal, "A Framework for Clustering Massive-Domain Data Streams," in *Data Engineering, 2009. ICDE '09. IEEE 25th International Conference on*, 2009, pp. 102-113.

[30] C. Aggarwal and P. Yu, "On clustering massive text and categorical data streams," *Knowledge and Information Systems*, vol. 24, pp. 171-196, 2009.

[31] C. C. Aggarwal and P. S. Yu, "A Framework for Clustering Uncertain Data Streams," in *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, 2008, pp. 150-159.

[32] S. Guha, A. Meyerson, N. Mishra, R. Motwani, and L. O'Callaghan, "Clustering data streams: Theory and practice," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 15, pp. 515-528, 2003.

[33] Q. Tu, J. F. Lu, B. Yuan, J. B. Tang, and J. Y. Yang, "Density-based hierarchical clustering for streaming data," *Pattern Recognition Letters*, 2011.

[34] B. Pardeshi, D. Toshniwal, N. Meghanathan, B. K. Kaushik, and D. Nagamalai, "Hierarchical Clustering of Projected Data Streams Using Cluster Validity Index Advances in Computer Science and Information Technology." vol. 131: Springer Berlin Heidelberg, 2011, pp. 551-559.

[35] L. Tu and Y. Chen, "Stream data clustering based on grid density and attraction," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 3, p. 12, 2009.

[36] S. Lühr and M. Lazarescu, "Incremental clustering of dynamic data streams using connectivity based representative points," *Data & Knowledge Engineering*, vol. 68, pp. 1-27, 2009.

[37] J. Wei and P. Brice, "Data stream clustering and modeling using context-trees," in *Service Systems and Service Management, 2009. ICSSSM '09. 6th International Conference on*, 2009, pp. 932-937.

[38] K. Chen and L. Liu, "HE-Tree: a framework for detecting changes in clustering structure for categorical data streams," *The VLDB Journal*, vol. 18, pp. 1241-1260, 2009.

[39] Guha, Meyerson, A. Mishra, N. Motwani, and O. C. . "Clustering data streams: Theory and practice ." *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, pp. 515-528, 2003.

[40] A. Jain , M. Murty , and p. Flynn " Data clustering: A review.," *ACM Computing Surveys*, vol. 31, pp. 264-323, 1999.

[41] I. Kononenko and M. Kukar, *machin learning and data mining*. Chichester, UK: Horwood Publishing, 2007.

Towards Processing Multi-Dimensional Dynamic Data

Yong Shi, Brian Graham
Department of Computer Science and Information Systems
Kennesaw State University
1000 Chastain Road
Kennesaw, GA 30144
yshi5@kennesaw.edu

Abstract— *Cluster and outlier detection has always been one of data mining research interests. Numerous approaches have been designed to find clusters and detect outliers in various types of data sets. In this paper, we present our research on analyzing data sets with constant changes. We design approaches to keep track of status of clusters, the movement of data points, and the updated group of outliers. Different from the traditional approaches which are focused on two-dimensional or low-dimensional data spaces, we aim to analyze data sets in multi-dimensional data spaces. We also propose to adjust the clusters and outliers simultaneously, since they are two concepts that are closely related.*

1. Introduction

Everyday a large amount of real data sets are generated in many disciplines. Data mining approaches are designed to analyze those data sets. Cluster and outlier detection has always been one of the focuses of data mining research. Cluster analysis specializes in techniques for grouping similar objects into a cluster in which objects inside a cluster exhibit certain degree of similarities, and separates dissimilar objects into different clusters. It is a method of unsupervised learning, and a common technique for statistical data analysis used in many fields, including machine learning, data mining, pattern recognition, image analysis and bioinformatics. Existing clustering algorithms can be broadly classified into four types: partitioning [10], [11], [12], hierarchical [21], [7], [8], grid-based [18], [15], [3], and density-based [5], [9], [4] algorithms.

Partitioning algorithms construct a partition of a database of n objects into a set of K clusters, where K is an input parameter. In general, partitioning algorithms start with an initial partition and then use an iterative control strategy to optimize the quality of the clustering results by moving objects from one group to another. Hierarchical algorithms create a hierarchical decomposition of the given data set of data objects. The hierarchical decomposition is represented by a tree structure, called dendrogram. Grid-based algorithms quantize the space into a finite number of grids and perform all operations on this quantized space. These approaches have the advantage of fast processing time independent of the data set size and are dependent only on the number of segments in each dimension in the quantized space. Density-based approaches are designed to discover clusters of arbitrary shapes. These approaches hold that, for each point within a cluster, the neighborhood of a given radius must exceed a defined threshold. Density-based approaches can also filter out outliers.

Each of the existing clustering algorithms has both advantages and disadvantages. The most common problem is rapid degeneration of performance with increasing dimensions [9], particularly with approaches originally designed for low-dimensional data. To solve the high-dimensional clustering problem, dimension reduction methods [3], [2], [14] have been proposed which assume that clusters are located in a low-dimensional subspace.

An outlier is a data point that does not follow the main characteristics of the input data. Outlier detection is concerned with discovering the exceptional behaviors of certain objects. It is an important branch in the field of data mining with numerous applications, including credit card fraud detection, discovery of criminal activities, discovery of computer intrusion, etc. In many applications outlier detection is at least as significant as cluster detection. There are numerous studies on outlier detection [19], [17], [20], [13].

In this paper, we analyze data sets with constant changes. We design approaches to keep track of status of clusters, the movement of data points, and the updated group of outliers. Different from the traditional approaches which are focused on two-dimensional or low-dimensional data spaces, we aim to analyze data sets in multi-dimensional data spaces. We also propose to adjust the clusters and outliers simultaneously, since they are two concepts that are closely related.

2. Related Work

Numerous approaches have been designed to analyze data sets with constant changes. For example, Abrantes etc. [1]

proposed a data clustering method that extends well-known static clustering algorithms, applying a motion model to track clusters that deform and translate. The method uses centroids, or points of reference within the data cluster that are drawn towards the center of clusters. The clusters are defined by their relevance to the centroids. In every instance of tracking the previous centroids are used to calculate translations and deformations of the cluster, and establish new centroids, based on previous calculations. They also demonstrated examples of this process in the context of object tracking using pixels and three dimensional linear calculations.

Garcia etc [6] proposed a method for clustering data with a dissimilarity measure and a dynamic procedure of splitting, giving examples of the method by using plot graphs of two-dimensional data sets. The dissimilarity measure uses the optimum path between each successive datum. The optimum path is chosen by finding the shortest distance between two successive vertices. This measure allows the clusters to take a unique shape, rather than clustered into quadrants. The optimal partitioning can be performed using the previous optimum path as a comparison, so clusters that are dense will be less likely to assume outliers or data belonging to another cluster. The authors also described a method to scale and smooth the derivatives.

Our approach is different from the previous work in that, instead of solely focusing on clustering analysis, we keep track of the change of clusters and outliers, and always keep them in the most updated status. The characteristic of certain data points in clusters and certain outliers are also changed dynamically. Furthermore, we design our approach in multi-dimensional data spaces instead of two-dimensional or low-dimensional data spaces.

3. Analyzing Dynamic Data Sets

A lot of algorithms have been designed for cluster analysis and outlier detection. It is difficult to detect clusters and outlier with a high accuracy for multi-dimensional noisy data sets, especially when the data sets change constantly. For example, Figure 1 shows a two-dimensional data set whose data points move dynamically over the time. The directions of the movement for certain data points are unpredictable.

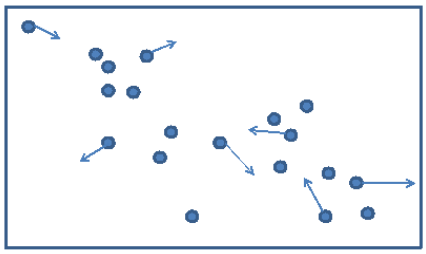


Figure 1: An example of dynamic data set

In this section we discuss how to design an approach to keep track of the status of clusters, the movement of data points, and the updated group of outliers. In order to describe our approaches, we shall introduce a few notations and definitions. Let *n* denote the total number of data points and *d* be the dimensionality of the data space. Let D_k be the *k*th dimension, where k = 1, 2, ..., d. Let the input *d*-dimensional data set be

$$\mathbf{DS} = \{X_1, X_2, ..., X_n\},$$

which is normalized to be within the hypercube $[0, 1]^d \subset R^d$. Each data point X_i is a *d*-dimensional vector:

$$X_i = [x_{i1}, x_{i2}, ..., x_{id}]. \quad (1)$$

Our approach is designed to keep track of the variation of clusters and outliers for a give data set with multiple dimensions. For a given step, let the current number of clusters be k_c and the current number of outliers be k_o; let the set of clusters be $\mathcal{C} = \{C_1, C_2, ..., C_{k_c}\}$, and the set of outliers be $\mathcal{O} = \{O_1, O_2, ..., O_{k_o}\}$.

For each cluster $C_i \in \mathcal{C}$, i=1,2,...,k_c, we define its size. For data sets in a two-dimensional space, a traditional way is to use the radius to represent how large is cluster is:

$$radius(C_i) = \max_{X_p \in C_i} (d(X_p, m_{c_i})), \quad (2)$$

where m_{c_i} is the centroid of Cluster C_i, X_p is any data point in Cluster C_i, and $d(X_p, m_{c_i})$ is the distance between X_p and m_{c_i} under certain distance metric, normally Euclidean distance for two-dimensional data space.

However, as the dimensionality of the data space goes higher, the radius of a cluster will increase dramatically. This is because as shown in equation 2, the distance between a data point in a cluster and its centroid is calculated by $d(X_p, m_{c_i})$ using Euclidean distance, and it is well known that Euclidean distance increases very fast when the dimensionality goes higher. This is called the "curse of dimensionality".

We can also analyze the case in another way. In a two-dimensional data space, a cluster is represented by a circle. The volume of the circle for a cluster can be calculated as πr^2, where r is the radius of the cluster. In a multi-dimensional data space, a cluster will be represented by a hyper-sphere S. There may be many empty regions which contain no data, and the bounding hyper-spheres of two different clusters may overlap. The volume v of the hyper-sphere S in a d-dimensional data space is calculated as

$$v = \frac{2\pi^{d/2} r^d}{d\Gamma(\frac{d}{2})}, \quad (3)$$

The gamma function $\Gamma(x)$ is defined as:

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt, \quad (4)$$

where $\Gamma(x+1) = x\Gamma(x)$ and $\Gamma(1) = 1$.

From equation 3 and equation 4 we can see that the volume v of the hyper-sphere S for a cluster increases dramatically as dimensionality goes higher.

Here we apply a different approach to define the size of a cluster. Let D_k be the *k*th dimension, where k = 1, 2, ..., d, the lower bound of the value range on D_k be the smallest value of the data points on D_k, and the upper bound of the value range on D_k be the largest value of the data points on D_k. For a given data set DS in a *d*-dimensional data space, we represent the size of a cluster C in DS as a group of intervals:

$$Size(C) = \{[l_1, h_1], [l_2, h_2], ..., [l_d, h_d]\}, \quad (5)$$

where l_k and h_k are the lower bound and upper bound of the value range on D_k , k = 1, 2, ..., d. The reason we define the size of a cluster C in this way is that, when the dimensionality goes higher, the size of C will not increase dramatically like what the Euclidean distance causes. Instead, there will be just more pairs of lower bound and upper bound added in the size of the C.

In our approach, we closely keep track of the change of clusters and outliers based on the movement of the data points in a data set DS. Based on how fast the data points in DS change their positions and availabilities, a time interval t is assigned to DS.

At each time interval t:

1) For each cluster $C_i \in \mathcal{C}$, we check the change of position for each data point $X_p \in C_i$, and count the number n_i of data points in C_i whose new value(s) on a certain dimension or certain dimensions are out of the value intervals defined in $\{[l_1, h_1], [l_2, h_2], ..., [l_d, h_d]\}$. If a data point no longer exists in DS, i.e., it is deleted from DS, it will also be counted into n_i.

If n_i exceeds a certain threshold, we will modify the intervals in $\{[l_1, h_1], [l_2, h_2], ..., [l_d, h_d]\}$ so it will still contain those data points, because in this case the majority of data points in C_i are moving out of the range of C_i, thus the size and shape of C_i need to be adjusted to still form a valid cluster. If n_i does not exceeds the threshold, which means those data points are the minority in C_i, they should be removed from C_i. For each of those data points, we will check to see if it resides in the new range of other clusters in $\mathcal{C}$. If it does, we will assign it to the new cluster, otherwise, we will assign it as a new outlier.

2) For each outlier O_j in $\mathcal{O}$, we will check to see if the new position of O_j resides in the new range of a cluster C_q in $\mathcal{C}$. If it does, we will assign O_j as a new data point in C_q, otherwise, O_j remains as an outlier.

The dynamic cluster-outlier adjustment algorithm is described in figure 2.

Algorithm: Dynamic data set process
Begin

a) Generate the groups of clusters and outliers from the initial data set $\mathbf{DS} = \{X_1, X_2, ..., X_n\}$. Various algorithms can be adopted to perform the initial clustering and outlier detection step, such as [16], etc.

b) Define a time interval t based on the frequency of change for positions and availabilities of data points in DS.

c) Monitor and record the change of data points' positions and availabilities dynamically.

d) At each interval t, perform step 1) and 2) mentioned in the last subsection.

e) Keep performing the algorithm until the data set no longer changes or the user interrupts the process.

f) Output the currently updated data set DS, the current set of clusters $\mathcal{C} = \{C_1, C_2, ..., C_{k_c}\}$, and the current set of outliers $\mathcal{O} = \{O_1, O_2, ..., O_{k_o}\}$.

End.

Figure 2: Algorithm: Dynamic Data Set Process

3.1 Time and space analysis

Suppose the size of the data set is n and the dimensionality is *d*. Throughout the process, in each time interval, we need to keep track of the change of values of all points, which collectively occupies $O(dn)$ space.

In each time interval, and for each cluster, we need to calculate the value n_i which is the number of data points whose values are out of the value intervals of its cluster. The time required for this process is $O(dn)$. Suppose there are T_n intervals before the algorithm is terminated. The processing time is $O(T_n dn)$.

4. Experiments

Various experiments were performed to evaluate and demonstrate the effectiveness and efficiency of the proposed approach. Our experiments were run on Intel(R) Pentium(R) 4 with CPU of 3.39GHz and Ram of 0.99 GB.

A synthetic data generator was generated to test the scalability of our algorithm over data size, dimensionality and time intervals. It produces data sets with normalized distributions. The sizes of the data sets vary from 2,500, 5,000, ... to 20,000, with the gap of 2,500 between each two adjacent data set sizes, and the dimensions of the data sets vary from 5, 10, ... to 40, with the gap of 5 between each two adjacent numbers of dimensions. To simulate the dynamic change of the data set, we applied the following strategies: 1) A time trigger was designed; 2) Every time the time trigger is randomly turned on: 2.1) A random subset RA of data points

from the data set are selected to have their values changed, 2.2) A random subset RB of data points are removed from the data set, 2.3) A set RC of randomly generated data points are inserted into the data set. With these steps the data sets are constantly changing.

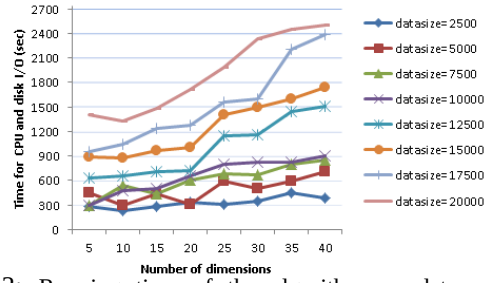


Figure 3: Running time of the algorithm on data sets with increasing dimensions

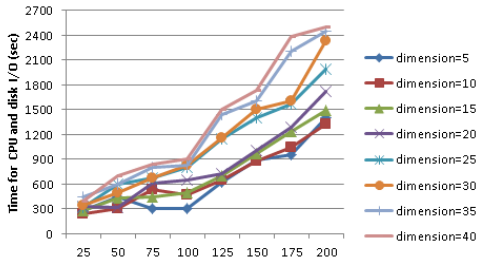


Figure 4: Running time of the algorithm on data sets with increasing sizes

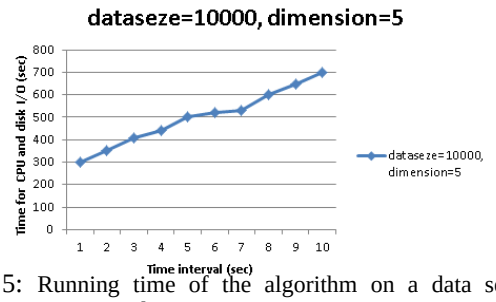


Figure 5: Running time of the algorithm on a data set with increasing time intervals

Figure 3 shows the running time of groups of data sets with dimensions increasing from 5 to 40. Each group has a fixed data size (from 2,500, 5,000, ... to 20,000). And we set the time interval as 1 second.

Figure 4 shows the running time of groups of data sets with sizes increasing from 2,500 to 20,000. Each group has fixed number of dimensions (from 5, 10, ... to 40). And we set the time interval as 1 second. The two figures indicate that our algorithm is scalable over dimensionality and data size.

Figure 5 shows the running time of a data set with 10000 data points and 5 dimensions. The time interval changes from 1 to 10 seconds, with the gap of 1 second between each two adjacent time intervals. Figure 5 indicates that our algorithm is scalable over the time intervals.

5. Conclusion and discussion

In this paper, we present a novel approach to analyzing the dynamic multi-dimensional data sets, which always keeps the clusters and outlier in the most updated status when the data points in the data sets change their positions and availabilities constantly. We will further conduct more experiments on synthetic and real data sets to test and demonstrate the efficiency and effectiveness of our approach.

References

[1] A. J. Abrantes and J. S. Marques. A method for dynamic clustering of data. In *BMVC*, 1998.

[2] C. C. Aggarwal, C. Procopiuc, J. Wolf, P. Yu, and J. Park. Fast algorithms for projected clustering. In *Proceedings of the ACM SIGMOD CONFERENCE on Management of Data*, pages 61–72, Philadelphia, PA, 1999.

[3] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, pages 94–105, Seattle, WA, 1998.

[4] Ankerst M., Breunig M. M., Kriegel H.-P., Sander J. OPTICS: Ordering Points To Identify the Clustering Structure. *Proc. ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'99), Philadelphia, PA*, pages 49–60, 1999.

[5] M. Ester, K. H.-P., J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, 1996.

[6] J. A. García, J. Fdez-Valdivia, F. J. Cortijo, and R. Molina. A dynamic approach for clustering data. *Signal Process.*, 44:181–196, June 1995.

[7] S. Guha, R. Rastogi, and K. Shim. Cure: An efficient clustering algorithm for large databases. In *Proceedings of the ACM SIGMOD conference on Management of Data*, pages 73–84, Seattle, WA, 1998.

[8] S. Guha, R. Rastogi, and K. Shim. Rock: A robust clustering algorithm for categorical attributes. In *Proceedings of the IEEE Conference on Data Engineering*, 1999.

[9] A. Hinneburg and D. A.Keim. An efficient approach to clustering in large multimedia databases with noise. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, pages 58–65, New York, August 1998.

[10] J. MacQueen. Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. Volume I,Statistics.*, 1967.

[11] L. Kaufman and P. J. Rousseeuw. Finding groups in data: an introduction to cluster analysis. 1990.

[12] R. T. Ng and J. Han. Efficient and Effective Clustering Methods for Spatial Data Mining. In *Proceedings of the 20th VLDB Conference*, pages 144–155, Santiago, Chile, 1994.

[13] G. L. Peterson and B. T. McBride. The importance of generalizability for anomaly detection. *Knowl. Inf. Syst.*, 14(3):377–392, 2008.

[14] T. Seidl and H. Kriegel. Optimal multi-step k-nearest neighbor search. In *Proceedings of the ACM SIGMOD conference on Management of Data*, pages 154–164, Seattle, WA, 1998.

[15] G. Sheikholeslami, S. Chatterjee, and A. Zhang. Wavecluster: A multi-resolution clustering approach for very large spatial databases. In *Proceedings of the 24th International Conference on Very Large Data Bases*, 1998.

[16] Y. Shi and A. Zhang. Towards exploring interactive relationship between clusters and outliers in multi-dimensional data analysis. In *International Conference on Data Engineering (ICDE)*, 2005.

[17] Y. Tao, X. Xiao, and S. Zhou. Mining distance-based outliers from large databases in any metric space. In *KDD*, pages 394–403, 2006.

[18] W. Wang, J. Yang, and R. Muntz. STING: A Statistical Information Grid Approach to Spatial Data Mining. In *Proceedings of the 23rd VLDB Conference*, pages 186–195, Athens, Greece, 1997.

[19] M. Wu and C. Jermaine. Outlier detection by sampling with accuracy guarantees. In *KDD*, pages 767–772, 2006.

[20] J. Yang, N. Zhong, Y. Yao, and J. Wang. Local peculiarity factor and its application in outlier detection. In *KDD*, pages 776–784, 2008.

[21] T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: An Efficient Data Clustering Method for Very Large Databases. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, pages 103–114, Montreal, Canada, 1996.

SESSION

REGRESSION AND CLASSIFICATION + FEATURE SELECTION

Chair(s)

Dr. Robert Stahlbock
Dr. Gary M. Weiss

MineTool-M^2: An Algorithm for Data Mining of 2D Simulation Data

Tamara B. Sipes and Homa Karimabadi[1,2]
[1] University of California San Diego, La Jolla, CA
[2] SciberQuest, Inc., Del Mar, CA
tsipes@ucsd.edu, homa@eng.ucsd.edu

ABSTRACT
Extraction of knowledge from massive and complex data sets generated from peta-scale simulations poses a major obstacle to scientific progress. We propose a new approach to solving this problem by utilizing an innovative feature extraction technique in combination with specialized data mining algorithms which can be incorporated as part of the scientific visualization pipeline. In this paper we show how data from simulations as well as many other real life examples can be represented in a form of multivariate time series. Accordingly, we have adapted a multivariate time series analysis data mining technique to handle simulation data. The technique extracts global features and metafeatures in the 2D simulation dataset in order to capture the necessary time-lapse information. The features are then used to create a static, intermediate data set that is suitable for analysis using the standard supervised data mining techniques. The viability of the new algorithm called MineTool-M^2 is demonstrated through its application to the problem of automatic detection of flux transfer events (FTE) in the simulation data. MineTool- M^2 built model led to a high FTE classification model accuracy of 95.56% correctly classified instances where the model produced one of three outputs of non-FTE, across cut FTE, and tangent cut FTE. For comparison, two other means of treating the time series data including a common summary statistics technique yielded much lower accuracies of 48% and 62% correctly classified instances, illustrating the complexity of this problem and the need for advanced techniques to handle such data.

Keywords
Multimedia Mining, Temporal and Spatial Data Mining, Multivariate Time Series Classification, Regression/Classification

1. INTRODUCTION
Scientific simulations have been used in a variety of fields to aid the understanding of a variety of scientific processes and enable scientific discovery. In space sciences for example, where progress relies on use of computer simulations in close ties with *in situ* and remote spacecraft measurements, the data challenge is particularly acute and has reached a critical stage. The advent of petascale computing has led to a significant increase in the size of the simulations. Our largest simulations include over 3.2 trillion particles, and 15 billion cells, and are run for several days using 200 K cores on Jaguar. We achieve about 7-9 million particle pushes per core for Cray machines. **Knowledge discovery from these increasingly complex and large data sets is a major bottleneck to progress in a variety of scientific fields today.** There is an eminent need for automated, intelligent methods to enable analysis and knowledge discovery in simulation data.

There are at least a couple of ways one can analyze data utilizing automated approaches and avoiding the time-consuming and error-prone human eye in tracking an event in large simulation data repositories. One obvious way would be to think of a simulation as a series of images, and analyze a 'time series' of image data. This approach would entail an image representation that would encompass the important areas of the image, and presenting it in a series. Another way would be to concentrate on the particular area of the simulation that is of interest and focus on the features being created and changed in time. We adopted the later approach, as it decreases the complexity of the problem, to be able to emphasize the creation and evolution of the events in the simulation, in order to describe them, create a predictive model and obtain the ability to classify them. Our approach entails collecting a certain spatial and temporal information, or features of the event in the simulation window (as in a series of point coordinate values (x,y)), in addition to the other variables available, that describe the (x,y) simulated measurements. These features, or set of points being tracked over time, in effect add another dimension to the time series data at the input. In the sections below we describe how we devise and collect the simulation features as the series of data points, or "cuts" in the example simulation domain, and utilize intelligent data mining classification tools to extract knowledge from them.

In the recent years we introduced a technique called MineTool [10] with distinct advantages over standard data mining techniques. Besides offering high accuracy of the resulting predictive models, a key advantage of MineTool-like approach is that it makes data mining more accessible, by offering a self-contained step by step procedure for model building. MineTool was created to handle static (non-time series) data and further expanded to a multivariate time series analysis technique which is naturally incorporated into the MineTool modeling process, suitable for time series data analysis. Some of the immediate applications of the resulting method, called MineTool-TS (for MineTool-TimeSeries), include multiple event detection and event classification [11].

In this paper, we adapt MineTool-TS to handle simulation data and illustrate its pattern recognition capabilities applied to simulation data. The paper is organized as follows. Section 2 discusses the simulations; Section 3 discusses the time series analysis and the underlying algorithm of MineTool-TS. Section 4 describes the application to simulation data. Summary and discussion are presented in Section 5.

2. SIMULATION DATA
The example that we consider here is the 2D global hybrid simulations (where electrons are modeled as fluid particles, and ions as fully kinetic) of the Earth's magnetosphere [8][9] where interaction of the solar wind plasma and magnetic fields impinging on the Earth's dipole field is modeled. The simulations

are 2D in a sense that only spatial variations of the parameters in two dimensions are retained but all three components of the vectors such as the magnetic field are kept. One feature of particular interest is the so-called flux transfer events [5] which were first observed in spacecraft data and are thought to be magnetic flux ropes formed at the Earth's magnetopause. Many details regarding the FTEs remain poorly understood but peta-scale simulations are enabling us to finally settle many questions regarding their formation, structure, and evolution. Figure 1 shows several examples of FTEs in a 2D global simulation. The simulation box is 2000 x 2000 ion skin depths or about 20 earth radii in each direction. The size of FTEs is small compared to the overall size of the magnetosphere and they appear as regions with density enhancements in this figure. FTEs have complex structures in velocity and magnetic field (not shown). Simulations have one major advantage to spacecraft observations in that one has in effect a very good spatial coverage of FTE at any given time and can track its evolution in time. In contrast, a single spacecraft or even four-spacecraft as in the case of Cluster mission, have limited spatial coverage. Figure 1 shows three sample spacecraft trajectories. Our goal in this particular study was to determine whether data mining algorithms can distinguish between these different cuts which include cuts scheming the surface of the FTE (cut-A), across an FTE (cut-B), or cuts away from FTEs (cut-C). If successful, this would imply that data mining algorithms can equally be applied to spacecraft data to distinguish among these three cuts. It would also imply that there are distinct features among the variables that, for example, would enable the algorithm to distinguish between cuts across and along an FTE.

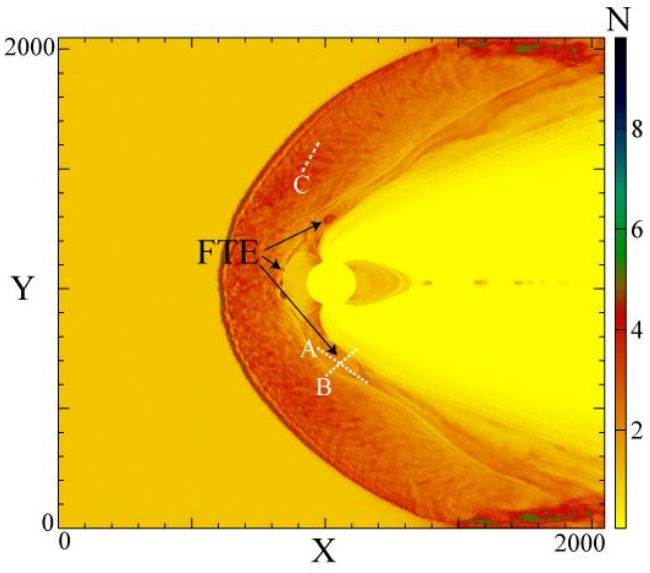


Figure 1. 2D simulation of the Earth's magnetosphere showing three examples of FTEs, and the three sample spacecraft trajectories (A, B and C).

3. TIME SERIES DATA ANALYSIS

3.1 Multivariate Time Series Data

In time series forecasting, one is interested in deciphering and quantifying temporal patterns in the data. In multi-variate time series data analysis, the relationship among the variables, each represented by a time series, can also be important. Time series analysis has become one of the most important branches of mathematical statistics and data mining, and a variety of techniques have been developed. The techniques range from a single time series forecasting (e.g., using the ARIMA method), to time series modification to allow certain patterns to be observed more easily (e.g., using FFT in signal processing), to multivariate time series classification. The latter is the focus of our work presented here.

3.2 Multivariate Time Series Classification

A data mining technique called MineTool-TS was introduced which captures the time-lapse information in multivariate time series data through extraction of global features and metafeatures [11]. In this paper we expand MineTool to handle not only static and time series data, but image, and simulation data as well, and call it MineTool-M^2 for MineTool-MultiMedia.

Time series data containing multiple variables (i.e. multivariate time series data) commonly occurs in a wide variety of fields including biology, finance, science and engineering. A time series (or more generally temporal data) is a sequence of measurements that follow non-random orders and can be generated either from a fixed point measurements at several time intervals or a convolved spatial-temporal variations as measured from a moving detector. Multivariate time series analysis is used when one wants to model and explain the interactions among a group of time series variables such as the field and plasma variables in the space physics domain. Much of the scientific data is in form of multivariate time series. Examples include ECG measurements, *in situ* field and plasma measurements of bow shock crossings, flux transfer events, turbulence in the solar wind, sign language hand movements, among others.

Multivariate time series classification attempts at classification of a new time series based on past observations of time series examples, rather than providing an analysis of a single-variate time series. Just like in any other classification problem, we are given examples of labeled data in order to build a predictive model. Historically, Hidden Markov Models (HMMs), recurrent Artificial Neural Networks (recurrent ANNs) and Dynamic Time Warping (DTW) have been used to build predictive models of multivariate time series data for classification tasks [15][21][24]. Even though these techniques are useful for certain tasks, they have several disadvantages which make them impractical for large datasets. In case of HMMs, for example, the number of parameters that needs to be set and examined is very large, even for small HMMs, determining the number of states for a certain dataset is just an educated guess, leading to many iterations of examining and setting the parameters. HHMs also do not handle continuous values very well, and make several major assumptions not readily available in a real-world scientific dataset. Recurrent ANNs suffer from several of the same problems as HMMs and require the user to experiment and choose many parameters and decide on the appropriate network architecture. The result is also in the form of a black-box which makes it difficult to understand.

If one could replace the time series by a static data consisting of variables that capture the relevant and interesting features (e.g., number of zero crossings, slope, extreme values) of the time series, then the standard MineTool technique could be used. Two

ideas for reduction of time series data immediately come to mind. First, one can randomly select several time instances of the data and treat each instance as a static data. The number of instances selected can be smaller than the total number of time instances available. Second, one can create summary statistics data, i.e., the time series data is replaced by its statistical measures such as the mean, standard deviation, minimum and maximum values, etc. As we will show shortly, even though these techniques are somewhat successful for a small number of simple datasets and problems, neither of these two approaches yields high accuracy results in modeling real life, complex time series data. Instead we use a more sophisticated approach to extract features from multivariate time series data that yields much higher accuracy [7][11].

3.2.1 MineTool for Static Data

The core data mining algorithm that underlies MineTool-TS is MineTool [10]. The advantages of MineTool over traditional algorithms such as support vector machine and artificial neural net (ANN) are its automated steps that make it more accessible and applicable in a variety of domains, accuracy, robustness and the analytical form of the model at the output.

An important algorithmic issue in data mining is how to find the optimal complexity of the model or the fitting function. Too much complexity in the model can result in overfit, whereas not enough complexity can result in underfit. The mathematical foundations of MineTool are based on considerations to balance the competing dangers of underfit and overfit to identify the level of model complexity that guarantees the best out-of-sample prediction performance without ad hoc modifications to the fitting algorithms themselves [14][17][18][26]. MineTool creates a predictive model architecture that is linear in the parameters. The algorithm searches for a model M that best relates rows of the input variable values Xij to the appropriate target value yi (yi = M(Xij)), where i = 1,…,N and j = 1,…,K. The model parameters are either linear combinations of the input ($\mathbf{X_i}$**'α**, where prime indicates transpose of the vector, index *i* refers to the i^{th} observation**)**, linear transformations of the input variables (**ζ*(Xi)***), or highly non-linear transformations of the input (Ψ**(Xi,γ)**). Equation 1 describes the general form of a MineTool model:

$$y_i = \mathbf{X_i'}\boldsymbol{\alpha} + \sum_{p=1}^{P} \zeta(\mathbf{X_i})'\boldsymbol{\delta_p} + \sum_{q=1}^{Q} \boldsymbol{\Psi}(\mathbf{X_i}, \boldsymbol{\gamma_q})'\boldsymbol{\beta_q} \qquad (1)$$

In its simplest form, the model would be a linear combination of the input parameters (i.e. a linear regression model). MineTool goes beyond a simple linear model by introducing the linear (such as level-1 and level-2 transformations producing cross-products, ratios, squares, cubes etc.) and non-linear transformation of the input variables, if their addition increases the model accuracy. The non-linear transforms Ψ are single hidden layer feed forward Artificial Neural Net (ANN)-like transforms, just like the ANNs of the same architecture, with the difference that the non-linear transformed inputs are combined into a linear model.

3.2.2 Metafeature and Global Feature Detection

To be able to process a (time) series dataset (represented with multiple rows of data describing one instance or observation) using MineTool, the data needs to be "flattened," or made static. Nevertheless, this needs to be accomplished without losing the important information incorporated in sequential measurements varying with time. Historically, this has been done either by summarizing the data and writing only the mean of the different row values of one observation, or recording the difference between the pairs of rows and then treating them as single instance entries. These techniques work somewhat well on just a limited set of time series problems. For real life, complex scientific datasets, these approaches are most often too weak to incorporate the important time changes in the data. The MineTool-TS solution to this problem is to collect the important time-changing information that can occur in one of the time series variables. While a value varies with time, it most often increases, decreases or stagnates. There are other, more complex features one can record, that consist of the three basic changes, such as bipolar signature (relevant in case of flux transfer events), where a value goes up, then goes down crossing the axis, and goes up again (the sinusoid function has a demonstrates the bipolar behavior, for example). Global features, just like the metafeatures, are used to extract the information from all the rows representing one observation. Global features describe one instance rows using one measurement, such as: the maximum value, minimum values, mean, mode or the number of zero crossings. Some of the metafeatures and global features included in the MineTool-TS algorithm are following:

- **Increasing Metafeature**— An increasing metafeature is recorded for all the consecutive rising time-series measurements. For each increasing event, we record its start point, duration, gradient and average value, so that the increasing events can be used for analysis and comparison.
- **Decreasing Metafeature**— A decreasing metafeature is recorded for all the consecutive reducing time-series measurements. For each decreasing event, similarly to the increasing events, we record starting point, duration, gradient (which is negative in this case) and average value.
- **Plateau Metafeature**— A plateau metafeature is recorded for all the consecutive non-changing time-series measurements. MineTool-TS allows for a small amount of noise to be ignored, so that the true plateaus are captured.
- **Bipolar Signature Metafeature**— A bipolar signature metafeature is recorded for all the consecutive time-series measurements that increase, decrease and cross the zero, and increase again.
- **Global Minimum**—For each single variable, the global minimum feature extracts the minimum value of all of the time observations belonging to one time series instance for the variable, and records it as the global minimum feature for that input channel.
- **Global Maximum**—The maximum value of all of the time observations belonging to one time series instance for the variable, and is recorded as the global maximum feature for that variable.
- **Mean** —The average value of all of the time observations belonging to one time series instance for the variable, and is recorded as the global mean feature for that specific variable.
- **Mode** —The mode value of all of the time observations belonging to one time series instance for the variable, and is recorded as the global mode feature for that specific input variable.
- **Number of Zero Crossings** —Lastly, the number of zero crossings occurring during the time observation recorded measurements is written down as the number of zero crossings global feature.

Next, once all the requested features are collected, the MineTool-TS algorithm performs the feature space segmentation to group

similar features and make them have a higher predictive value for data mining. More details on the algorithm can be found in [11].

3.3 MineTool-M^2 Extension for Multimedia Data Mining

The time series classification algorithm needed to be adapted to handle simulation (and other multimedia) data. Figure 2 illustrates the additions to the basic time series analysis algorithm:

(i) Multimedia (i.e. simulation) data preparation

(ii) Handling of the time series of uneven lengths

3.3.1 Simulation Data Preparation

The simulation data needs to be converted into a series dataset as the algorithm is designed for time series data. A dataset containing a different type of a series data could be entered at the input as well, as the metafeature variables would track the changes that occur either from one point in time to another (as in time series data) or from one point in space to another (as in spatial data). To prepare simulation data for being entered in MineTool-M^2 we perform a preprocessing step (Figure 2) that converts the multimedia data into a series data set. Section 4.2 details our feature extraction step that converts the simulation data into a series "cuts" data and enables further analysis.

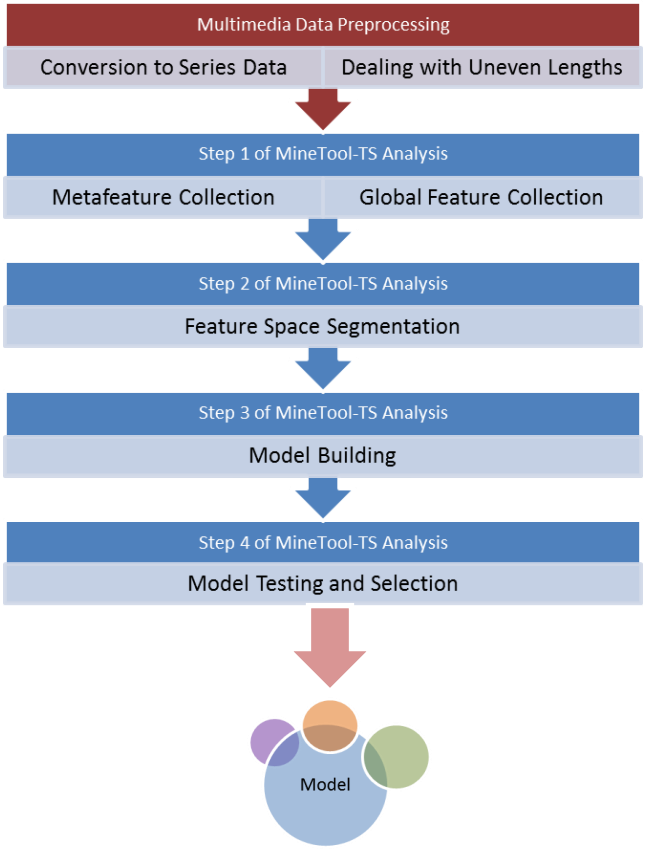


Figure 2. An Illustration of the MineTool-M^2 algorithm.

3.3.2 Time Series of Uneven Lengths

To be able to effectively describe a set of points being tracked in the simulation window, the algorithm needed to be able to handle uneven lengths of the series data. This means that one observation of interest could be tracking an event in the simulation from its creation to, for instance, the time = T_{117}, whereas the other event evolution could end at the time of T_{59}.

Therefore, we adapted the basic method to accept different series observation lengths resulting in MineTool-M^2 expecting an array at the input, listing the number of time observations (NTO) for each of the input series data instances. The original algorithm assumed that all of the series streams are of the equal lengths. Figure 2 illustrates the basic steps of the MineTool-M^2 method. First, the simulation data is converted into a multi-series form by adding a dimension to the expected time series data (resulting in extra columns of the input dataset). Since the data may or may not be of equal length, the algorithm expects an array of the series data lengths at the input. Consecutively, the multivariate time series classification steps of MineTool-TS are performed: metafeature and global feature collection, the feature space segmentation and reduction, following by the iterative model building and evaluation until the best model is selected.

In the following section we illustrate the application of MineTool-M^2 to the Flux Transfer Event (FTE) simulation data.

4. APPLICATION TO SIMULATION DATA

To demonstrate the applicability of MineTool-M^2 to mining time series multimedia data, we looked at the problem of automatic detection of Flux Transfer Events (FTE) in simulation data.

FTEs are typically identified on the basis of clear isolated bipolar signatures in the B_n component of the magnetic field (in the LMN coordinate system). The Cluster spacecraft magnetic field observations of 4-s resolution from the Fluxgate Magnetometer (FGM) [1] and plasma observations of 4-s resolution from the Cluster Ion Spectrometry (CIS) instrument [22] are commonly used for Cluster magnetopause crossings and FTE identifications. The measurements include a total of 11 input variables: B_x, B_y, B_z, $|B|$, N_p, V_x, V_y, V_z, $T_{\|}$, $T_{\perp}$,T_t However, simulation is used to enable visualization of what the collected measurements mean, how these events occur in magnetosphere, and aid the scientist in evaluating novel algorithms and gaining better understanding of these events.

4.1 Description of the Test Problem

FTEs are usually detected based on the data signatures tangent to the FTE events. The goal of the data analysis and modeling is to build a model that will be able to distinguish the cuts across the FTEs from the cuts tangent to FTEs (two classes), as well as differentiate non FTEs. This is a challenging three-class, multivariate data series classification problem. In FTE observations, scientists can identify FTEs only by looking at signatures tangent to FTEs and our goal is to, using simulation and the presented MineTool-M^2 approach to data mining of multimedia time series data, improve this fact.

4.2 Data Collection and Preparation: "Cuts" Feature Extraction

To analyze simulation data in tracking an event, we concentrate on the particular area of the simulations that is of interest and focus of these features being created and changing in time. In this manner, we are able to emphasize the FTE events in order to describe them, model and classify them. We introduce the *"cut" feature*, a novel computer vision feature extraction method that enables us to collect the important characteristic of the area of interest within simulation data window, while decreasing the complexity of the data selected for further analysis. A "cut" or a "slice feature" is a line drawn at the site of the feature of interest, or at the site of the feature non-existence. "Cuts" are modeled based on the *spacecraft trajectories* and, in effect, simulate what a spacecraft would observe while on a trajectory near an event or non-event. Our goal here was to determine whether data mining algorithms can distinguish between these different cuts. We have devised a cutting routine for making "cuts" or "slices" in the simulation data and creating a data file to be used in analysis and modeling. Figures 3a, 3b and 3c show three sample spacecraft trajectories-guided cuts or slices in the 2D simulation data which include cuts scheming the surface of the FTE (cut-A), across an FTE (cut-B), or cuts away from FTEs (cut-C). The variables that were observed in the cuts included:

X, Y, B_X_slice, B_Y_slice, B_Z_slice, Density_slice, T_{PAR}_slice, T_{PERP}_slice, T_{TOTAL}_slice, V_{IX}_slice, V_{IY}_slice, V_{IZ}_slice, B_{TOTAL}, event

The simulation FTE data has been labeled with three labels: a) cuts tangent the FTE, b) cuts across to the FTE, and c) non-events. The dataset consists of series data and does not have to have the same length. In this phase of the project, we collected 45 of each of the types of FTE events, giving 135 total events, or streams of data. Each of our events had up to 1000 data points representing one cut, however the length was varying. We have prepared the data and converted in the form suitable for mining using our MineTool-M^2 method for multivariate classification of multimedia time series data.

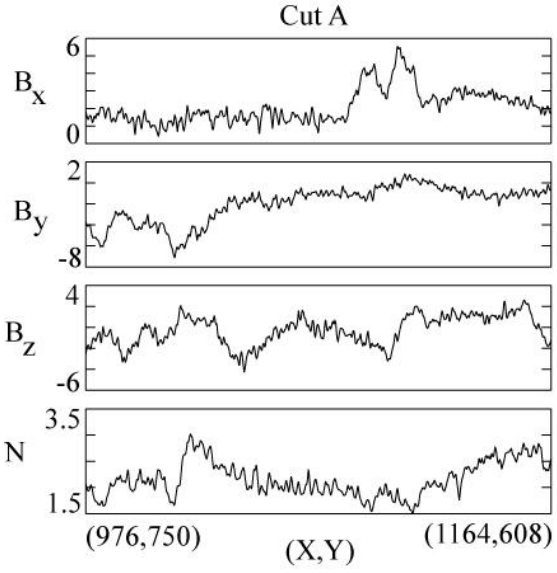


Figure 3a. A Cut in the Simulation Data Tangent to the FTE.

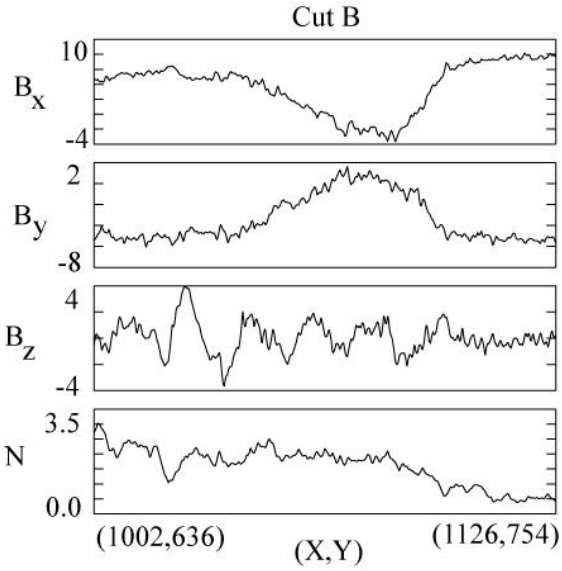


Figure 3b. A Cut in the Simulation Data Across the FTE.

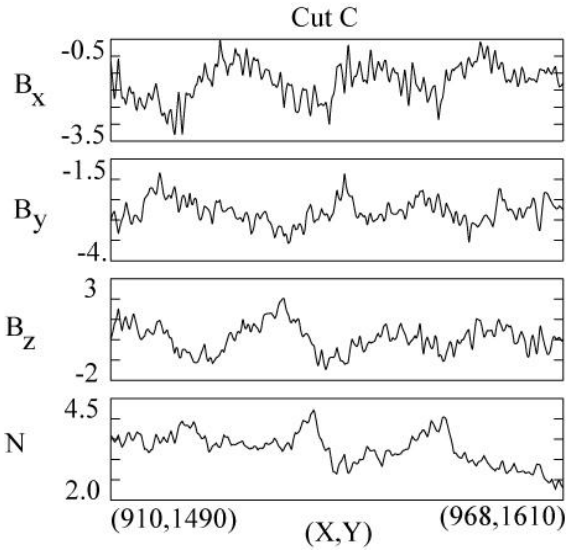


Figure 3a. A Cut in the Simulation Data Away From the FTE.

4.3 Modeling Results

Our method first converted the simulation cuts data into series data, followed by going through the series data and collecting the metafeature information, such as increases, decreases and plateaus in one series. Then, using this information each of the series was "flattened" into a static row of data and fed into the intermediate dataset. This was completed for each of the 135 event examples. The flattened, static dataset was then fed into our MineTool algorithm, to discover the correlations among the input variables to the output variables.

We contrast the modeling results of the flux transfer event (FTE) classification in simulation data performed in three different ways (as listed in Table 1): as a static dataset (where each row is treated

as an independent instance, and not as a part of a series), as a series data, using the summary statistics representation, where a series is converted into a single instance of data using measurements such as mean, standard deviation, minimum, maximum, range, number of zero crossings, interquartile range (or, the spread) and the median value, for each of the variables in the data), and as the true series data, using MineTool-M^2. The MineTool- M^2 approach requires the most computational time, followed by the summary statistics approach and static analysis approach. However, the accuracy of the model (96.6% vs. 62.4% vs 47.6%) is a worth-while trade off. Table 1 describes the results obtained in our study using standard data mining evaluation statistics (percentage of correctly classified instances, correlation coefficient, mean absolute error (MAE) and root mean squared error (RMSE)).

```
      event = 0.941734
      + 39.9488*Btotal_Inc_14*den_Inc_5
      -3.51784*vix_Dec_5
      -359.589*tperp_Dec_4*tperp_Dec_6
      -57.0447*tperp_Inc_2*viy_Dec_5
      -5.70272*tperp_Inc_3
      + 392.001*ttotal_Inc_10*ttotal_Dec_25
      + 53.5713*ttotal_Inc_20*viy_Inc_2
      -14.6957*tperp_Inc_2*tpar_Dec_4
      + 102.018*Btotal_Inc_10*vix_Dec_3
      + 95.3943*den_Inc_8*bx_Dec_22
      -61.8677*viz_Inc_9*ttotal_Dec_7
      -15.0103*Btotal_Inc_5*den_Dec_16

Where :
Btotal_Inc_14 represents a time series
feature with the following average
description:
      average value of -> 0.330258
      mid time value of -> 552.96
      gradient value of -> 0.002956
      duration of -> 64.3565

den_Inc_5 represents a time series feature
with the following average description:
      average value of -> 0.258179
      mid time value of -> 543.827
      gradient value of -> 0.00134285
      duration of -> 25.4539

vix_Dec_5 represents a time series feature
with the following average description:
       average value of -> 0.347282
      mid time value of -> 747.303
      gradient value of -> -0.00113456
      duration of -> 40.228  etc.
```

Figure 4. The Predictive Model of FTEs.

The modeling results are producing a model with 95.6% accuracy tested on a third of the data, set aside as holdout (test) data, and built on the 66% of the data as the training set, with each of the classes being equally represented in the training and test data. The model picks up on the most important metafeatures in the classification of an event as an across FTE, tangent FTE or non-event, and is given in Figure 4.

The predictive model created by the MineTool data mining method is in an analytical form, enabling insight into the most important metafeatures and global features detected by the algorithm in appropriately classifying a time series instance of data. The model in Figure 4 shows that the specific total magnetic field (B_{TOTAL}) together with the specific increments in density (which is a level-1 cross product linear transformation $\zeta(Xi)$) from the Eq. 1) positively correlates to a series cut variable being classified as an FTE, while if the V_X_Dec_5 (a simple linear combination of the input variable X_i'α) is detected, it negatively correlates with an FTE event (there were no highly non-linear transformations $\Psi(Xi,\gamma)$ in the model chosen by the method). The model is also able to very accurately distinguish between an event label 1 and 2 (across and tangent FTE).

Table1. Comparative analysis of MineTool-M^2 vs. other methods.

Type of Analysis	Correctly Classified	Correlation coefficient	MAE	RMSE
Static Data Analysis	47.6%	0.376235	0.5682	0.6932
Summary Statistics Analysis	62.4%	0.554823	0.4951	0.6351
MineTool- M^2	95.6%	0.952676	0.1772	0.2532

5. SUMMARY AND DISCUSSION

In this paper we aim to contribute to the urgent need to understand and learn from the often massive, constantly increasing, complex, multimedia data, often collected or created in the form of simulation data, in an automated fashion.

We adapt our multivariate time series analysis data mining technique to handle simulation data. We extract the important information from the simulation data by introducing a novel computer vision feature extraction operator named "cuts" that collect the cuts-type of data in the simulation window. The cuts-data are then converted into a series data and input into MineTool-M^2 for analysis and modeling. We also expand the method to allow for uneven lengths of the series data at the input. The technique extracts global features and metafeatures in the 2D simulation dataset in order to capture the necessary time-lapse information. The features are then used to create a static, intermediate data set that is suitable for analysis using the standard supervised data mining techniques.

The capability of the new algorithm called MineTool-M^2 is demonstrated through its application to the problem of automatic detection of flux transfer events (FTE) in the simulation data. MineTool- M^2 built model led to a high FTE classification model accuracy of 95.56% correctly classified instances where the model produced one of three outputs of across cut FTE, tangent

cut FTE, and non-FTE. For comparison, two other means of treating the series data including a common summary statistics technique yielded much lower accuracies of 48% and 62% correctly classified instances, illustrating the imminent need for advanced techniques, such as MineTool-M^2, to handle such data.

Our future work will encompass the expansion of MineTool-M^2 to 3D simulation data, and other multimedia data as well. By applying and extending ideas from data mining, image and video processing, statistics, and pattern recognition, we are developing a new generation of computational tools and techniques that are being used to improve the way in which scientists extract useful information from data.

6. ACKNOWLEDGMENTS

This work was supported by a NASA SBIR and NNX11AC83 grant at SciberQuest, Inc., Simulations were performed on Kraken, a Cray XT5 system provided by the National Science Foundation at the National Institute for Computational Sciences, and on NASA' s Pleiades, which is provided by the NASA High-End Computing (HEC) Program.

7. REFERENCES

[1] Balogh A, Dunlop MW, Cowley SWH, Southwood DJ, Thomlinson JG, Glassmeier KH, Musmann G, Luhr H, Buchert S, Acuna MH, Fairfield DH, Slavin JA, Riedler W, Schwingenschuh K, Kivelson MG, The Cluster magnetic field investigation, *Space Sci. Rev*., 79, 65-91, 1997.

[2] Candes, E.. *Ridgelets: Theory and Applications*. PhD thesis, Stanford University, Department of Statistics, 1998.

[3] Cortes C. and Vapnik V. Support-Vector Networks, *Machine Learning, 20, 1995.*

[4] Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1-38.

[5] R. C. Elphic. Observations of Flux Transfer Events: A Review. the American Geophysical Union, 1995.

[6] Hartley, H. (1958). Maximum likelihood estimation from incomplete data. *Biometrics*, 14:174–194.

[7] Kadous, M. W. *Temporal Classification: Extending the Classification Paradigm to Multivariate Time Series*. PhD thesis, School of Computer Science & Engineering, University of New South Wales, 2002.

[8] Karimabadi, H., and J. Dorelli, H. X. Vu, B. Loring, Y. Omelchenko, Is quadrupole structure of out-of-plane magnetic field evidence of Hall reconnection?, to appear in *Modern Challenges in Nonlinear Plasma Physics*, editor D. Vassiliadis, AIP conference, 2010.

[9] Karimabadi, H., H. X. Vu, D. Krauss-Varban, Y. Omelchenko, Global Hybrid Simulations of the Earth's Magnetosphere, *Numerical Modeling of Space Plasma Flows: Astronum-2006*, vol. 359, 257, 2006.

[10] Karimabadi, H., Sipes, T. B., White, H., Marinucci, M., Dmitriev, A., Chao, L.K., Driscoll, J., Balac, N. (2007). Data Mining in Space Physics: 1. The MineTool Algorithm, *J. Geophys. Res*., 112, A11215, doi:10.1029/2006JA012136.

[11] Karimabadi, H., Sipes, T. B., Wang, Y., Lavraud, B. and Roberts, A. (2009). A new multivariate time series data analysis technique: Automated detection of flux transfer events using Cluster data, *J. Geophys. Res*., *Vol 114, A06216, doi:10.1029/2009JA014202, 2009*

[12] Lendasse, A., Lee, J. , de Bodt, E., Wertz, V. and Verleysen, M.. Approximation by Radial Basis Function Networks Application to Option Pricing. In Connectionist Approaches in Economics and Management Sciences, C. Lesage, M. Cottrell eds., Kluwer academic publish., 2003, pp. 203-214

[13] Looney, C. G., Pattern recognition using neural networks, Theory and algorithms for engineers and scientists, Oxford University Press, 1997.

[14] Marinucci, M., Automatic Prediction and Model Selection,Ph.D. Thesis, Departamento de Fundamentos del Analisis EconomicoII, Facultad de Ciencias Economicas, Universidad Complutense de Madrid 2007.

[15] Myers, C. S. and Rabiner, L. R. A comparative study of several dynamic time-warping algorithms for connected word recognition. The Bell System Technical Journal, 60(7):1389-1409, September 1981.

[16] McLachlan, G. and Krishnan, T. (1997). *The EM algorithm and extensions*. Wiley series in probability and statistics. John Wiley & Sons.

[17] Pérez-Amaral, T., Gallo, G. M. and White, H., A Flexible Tool for Model Building: the Relevant Transformation of the Inputs Network Approach (RETINA), *Oxford Bulletin of Economics and Statistics*, 65 (s1), 821-838, 2003.

[18] Pérez-Amaral, T., Gallo, G. M. and White, H., A Comparison of Complementary Automatic Modeling Methods: RETINA and PcGets," *Econometric Theory*, 2005.

[19] Powell, M. J. D. *Radial basis functions for multivariate interpolation: A review*. In Algorithms for Approximation, J. C. Mason and M. G. Cox, Eds. Clarendon Press, Oxford, 1987.

[20] Ross Quinlan (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA.

[21] Rabiner, L. R. and Juang, B. H. An introduction to hidden markov models. IEEE Magazine on Acoustics, Speech and Signal Processing, 3(1):4-16, 1986.

[22] Reme, H. and C. Aoustin and J. M. Bosqued and I. Dandouras, B. Lavraud, et al., First multispacecraft ion measurements in and near the Earth's magnetosphere with the identical Cluster ion spectrometry (CIS) experiment, *Annales Geophysicae*, 19, 1303-1354, 2001.

[23] Ripley, B.D., Pattern Recognition and Neural Networks, Cambridge University Press; 1996.

[24] Schmidhuber, J., Graves, A., Gomez, F. and Hochreiter, S. Recurrent Neural Networks, Cambridge University Press, 2012.

[25] White, H., Approximate nonlinear forecasting methods, in *Handbook of Economic Forecasting*, Volume 1, Edited by Elliott, Granger and Timmermann, Elsevier, Amsterdam, 2006.

[26] White, H., Personnel Readiness: Neural Network Modeling of Performance-Based Estimates, *Final Report to the Office of Naval Research, Contract #: N00014-95-C-1078*, 1999.

[27] R. J. Vidmar. (1992, August). On the use of atmospheric plasmas as electromagnetic reflectors. *IEEE Trans. Plasma Sci.* [Online]. *21(3).* pp. 876—880. Available: http://www.halcyon.com/pub/journals/ 21ps03-vidma

Hybrid Predictive Models for Optimizing Marketing Banner Ad Campaign in On-line Social Network

M. Łapczyński[1] **and J. Surma**[2]
[1] Department of Marketing Research, Cracow University of Economics, Cracow, Poland
[2] Faculty of Business Administration, Warsaw School of Economics, Warsaw, Poland

Abstract - *Promotional campaigns implemented in web-based social networks are growing in popularity due to an increasing number of users in virtual communities. A study concerning an advertising campaign in a popular social network is presented in this article. Identification of the profile of a group of people responding positively to a banner ad allows for an effective management of marketing communications. Unfortunately, a small number of users clicking on ads leads to a situation in which researchers have problems with heavily skewed datasets. This article attempts to build hybrid predictive models based on clustering algorithm and decision trees. The choice of these analytical tools was to ensure a clear interpretation of the model using a set of if-then rules instead of black boxes with a high predictive power.*

Keywords: social network, web advertising, class imbalanced problem, hybrid predictive models

1 Introduction

On-line social networks have generated great expectations in the context of their business value. The straightforward approach of their monetization is to apply web banners (banner ad) campaigns. This form of online advertising entails embedding an advertisement into a web page, and the advertisement is constructed from an image. When viewers click on the banner, they are directed (click-through) to the website advertised in the banner. According to the latest marketing research customers actively avoid looking at online banner ads [1] and response rates to banner ads have fallen dramatically over time [2]. On the other hand, banner based advertisement campaigns on on-line social networks might be monitored in real-time and may be targeted in a comprehensive way depending on the viewers' interests. On-line social network users are identified by a unique login and leave both declarative (self reported) and real behavioral data [3]. Access to behavioral data constitutes a particular competitive advantage of an online social network as compared to other web portals. In this research we would like to focus on this potential supremacy of behavioral data mining for marketing campaign management based on web banners.

The main research problem is to optimize a marketing banner ad campaign by targeting an appropriate user, and to maximize the response measure by the click-through rate (response rate). An empirical evaluation presented in this paper is based on a marketing ad campaign for a cosmetics company. The authors decided to built hybrid predictive models based on classification and regression trees (C&RT) algorithm and clustering algorithms. In addition to profiling users potentially interested in advertising the other major goal of research is overcoming class imbalance problem that very often occurs in such experiments.

The description of hybrid models and ensemble classifiers applied in analytical CRM (Customer Relationship Management) is presented in section II. In Section III there is a description of the variables used in the analysis where we focus on class imbalance problem, which constitutes here the biggest challenge for researchers. The authors discuss two main strategies applied in the case of highly skewed data, i.e. sampling techniques and cost sensitive learning. They also refer to the results of research conducted on the basis of the same dataset. In Section IV we present a scheme of construction of hybrid predictive models. A series of experiments in building an effective data mining model can be found in Section V. Finally, in Section VI the paper concludes with a summary of the experiments results.

2 Hybrid predictive models – literature review

Advertisements click prediction models have long been of interest to many researchers. Many of their papers refer to the advertisements placed on search engines. Richardson et al. [4] use logistic regression and a set of independent variables relating mainly to the searched objects. 81 independent variables were grouped into five categories: appearance, capture attention, reputation, landing page quality, and relevance. Wang and Chen [5] compared the models obtained by using conditional random fields (CRF), support vector machines, decisions trees and back-propagation neural networks. Their effort was focused on choosing an appropriate analytical tool and selecting the best set of independent variables, for which they used a random subspace, F-score, and information gain.

In the literature there are also numerous papers related to data mining in social networks. In one of them [6] the authors grouped the users into cohesive subgroups from social networks. In the next stage they estimated preferences of

people belonging to each subset that were treated here as the probability of choosing a particular product. Calculations were based on past transaction data. A similar approach can be found in [7], where users were grouped into the so-called quasi social-networks. Authors assumed that people visiting the same social networking websites, photography sites, non-professional blogs, etc. have similar preferences and comparable likelihood of purchasing particular products. The authors decided to use hybrid predictive models because, according to their best knowledge, this approach was not used in predicting clicks in social networks.

The term "hybrid predictive model" appears in marketing in the context of choice models. Ben Akiva et al. [8] proposed the so-called hybrid choice model, which integrates many types of discrete choice modelling methods, i.e. a random utility model with observable independent variables, a latent class model, and a latent variable model. When building predictive models in analytical CRM one often needs to use approaches that combine numerous tools of the same kind or several different analytical tools. In the literature there are two terms to describe such procedures. One is the so-called ensemble model (committee), which refers inter alia to the random forest [9] or boosted classification and regression trees based on bootstrap subsamples [10], [11]. Ensemble models are also built by combining classical statistical tools such as probit models [12], which were used to predict cross-selling.

Some authors [13] distinguish between the so-called within-algorithm ensemble (combination of results obtained with the use of the same analytical tool), and cross-algorithm ensemble (aggregation of results obtained with the use of different tools). They applied TreeNet and logistic regression in their predictive model that was built for cross-selling purposes.

The other term is 'hybrid models', which usually occurs in the context of combining different methods. These research works cover a wide range of areas related to the analytical CRM and database marketing. Some authors [14] combined results obtained from clustering algorithms (K-means, K-medoid, self organizing maps, fuzzy c-means and Balanced Iterative Reducing and Clustering using Hierarchies) with results obtained from decision tree (C5.0). Their goal was to predict customer churn. In churn prediction combining SOM with decision trees was also called 'two-stage classification' [15]. Authors divided the sample into 9 clusters and built a separate decision tree model for the clusters where the percentage of churners was relatively high.

An example of combining clustering algorithm with decision trees can also be found in literature [16], where continuous variables (time series) and discrete variables were analyzed by using K-means method and C4.5 algorithm. In turn, [17] built a hybrid model to predict cross-selling. In their approach they employed logistic regression, AdaBoostM1 algorithm and voting feature intervals (VFI).

Hybrid models were also used for bankruptcy prediction, where the authors combined genetic algorithms, fuzzy c-means and Multivariate Adaptive Regression Splines (MARS) [18]. Another hybrid predictive model in this research area was based on genetic algorithm and neural networks [19].

It is also noteworthy that there were attempts of combining decision trees with logit models. Combining CHAID algorithm with binomial logit model [20], and a few years later CART algorithm with logit models [21] can be regarded as the first attempts to build such a model.

3 Description of data and class imbalance problem

The data set comprised 81,584 cases and 111 variables. The dependent variable referred to the positive response of an internet user to an internet banner ad of a cosmetics company. The positive response should be understood as clicking on the banner that resulted in visiting the cosmetics company website. The set of continuous and discrete independent variables referred to the five main areas: on-line activity of internet users, interaction with other people within the website, expenses, installed games and declarative demographic variables (gender, age, education).

The response category number of the dependent variable was very small (207 observations, i.e. 0.25% of the entire data set), which causes the researcher to confront here the problem of highly skewed data. This is a common and very troublesome inconvenience that appears while building predictive models for relationship marketing purposes. The disproportion between the number of 'ones' and the number of 'zeros' (positive and negative categories) refers to the customer churn analysis, customer acquisition, cross-selling, and in other disciplines for fraud detection or medical diagnoses.

In general, there are two main approaches [22] how to deal with this problem. One is based on changing the structure of a learning sample (sampling techniques), while the other one pertains to cost-sensitive algorithms. For highly skewed data one can use sometimes the so-called one-class learning, especially when gathering information about a minority class is difficult, or when the investigated area itself is of imbalanced nature.

One can increase the number of cases belonging to the minority class while changing the structure of the learning sample. It is referred to as up-sampling (over-sampling), which can be realized randomly, directly, or by gathering synthetic cases [23]. It is also possible to reduce the size of the majority class, which is referred to as down-sampling (down-sizing, under-sampling). In the case when one of the methods of data structure modification is applied we speak about one-sided sampling technique, and when both methods are applied we speak about two-sided sampling technique.

Using previously gained experience and experiments with the construction of predictive models [24] the authors decided to employ under-sampling and two-sided sampling technique. In both cases it led to a situation where the proportions of classes in the learning sample were 10%-90%, while the proportions of classes in the test sample remained

unchanged. Detailed information on the size and structure of various sets of observations is given in Section V.

When taking into account cost-sensitive learning one can distinguish [25] a group of direct algorithms (e.g. ICET or cost-sensitive decision trees), and cost-sensitive meta-learning methods (e.g. MetaCost, CostSensitiveClassifier, Empirical Thresholding or cost-sensitive Naive Bayes). In general, the goal is to increase predictive accuracy by assigning different costs to different categories of dependent variables. The authors decided to use classification and regression trees algorithm (C&RT), since this method is one of the first that utilized misclassification costs and a priori probabilities. The additional advantage of this tool is that it also provides a set of clear rules describing a model, and is therefore comprehensible for managers.

In previous attempts of building the predictive model three algorithms were applied: C&RT, random forest (RF), and boosted classification trees. Despite the fact that the best results were achieved with RF, its biggest drawback was the lack of a clear interpretation of the model. Marketers very often need more than an effective black box with a high predictive power. They want to know the qualitative nature of the relationships between variables, which will enable them not only to efficiently select the target group, but also to more thoroughly understand the studied phenomenon. It is also worth noting that C&RT provided positive results from the financial point of view, and in certain combinations of classification costs, a priori probabilities and sampling techniques outperformed other tools. An additional advantage of this algorithm is the above mentioned ability to change misclassification costs and a priori probabilities of classes occurrence, which makes it potentially useful in solving the problem of imbalanced classes. All this resulted in the authors' decision to create hybrid models in which the fundamental role is played by C&RT.

4 Description of hybrid model

4.1 Hybridization

Authors treat building of a hybrid model as a sequential combination of supervised and unsupervised models. Another reason for naming this approach a "hybrid" is a combination of classical statistical tools (K-means method) with the algorithm derived from data mining (C&RT). In the first stage objects were clustered by using K-means algorithm and self-organizing maps (SOM), also known as Kohonen networks. In the second stage C&RT algorithm was applied, treating cluster membership of the objects as a new independent variable (model M1 and M3) and building different C&RT models for each cluster separately (model M2 and M4). Modeling procedure is shown in Fig. 1.

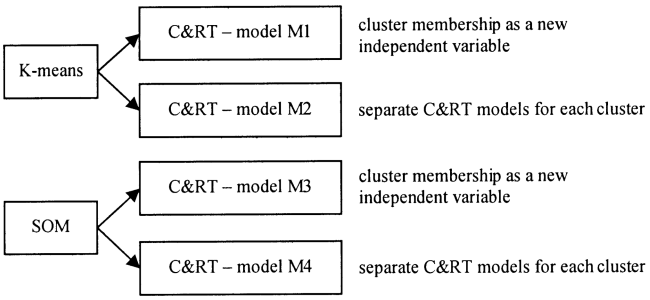


Fig. 1. The procedure for building hybrid models

4.2 Clustering by using K-means algorithm

According to one of the approaches to the procedure of building clusters [26], one should check the degree of correlation of candidate variables prior to the selection of variables. If two of them were highly correlated with each other, one of them should be removed from the analysis. Therefore, prior to the analysis we reduced the number of 46 variables using the principal component analysis. The purpose of PCA is to reduce the multidimensional space to a smaller number of uncorrelated principal components.

The first principal component explains the highest percentage of overall variance. The second principal component achieves the highest percentage of variance of all remaining variables, etc. In determining the number of principal components the authors used the Kaiser's criterion, which states that the eigenvalue should be greater than one [27].

K-means algorithm is sensitive to differences in variables' units and ranges. Standardizing, indexing or normalizing [28] are frequently used methods of rescaling variables into the same range. Gan, Ma, and Wu [29] mention various ways of standardizing variables that are based on mean, median, standard deviation, range, Huber's estimate, Tukey's biweight estimate, and Andrew's wave estimate. In this case, normalization was applied using a popular formula (Xi - Xmin) / (Xmax - Xmin), where Xmin represents the lowest and Xmax the highest value of a given variable.

Overall, the purpose of the analysis is to identify clusters that are maximally homogeneous, and at the same time differ among themselves. Therefore, the authors decided to calculate the between sum of squares (BSS) and the within sum of squares (WSS) for each set of clusters. The index of WSS/BSS subsequently allowed to choose the optimal number of clusters. It is often considered, however, that the final number of clusters is determined by practical reasons and the ability to use the analysis results in business activity.

4.3 Clustering by using Kohonen networks

Self-organizing maps (also referred to as Kohonen networks) are a variation of unsupervised neural networks and are considered as an alternative clustering method [30]. In general, similarly to neural networks self-organizing maps have the input layer and the output layer. Every case targets only one field of the topological map and has its own

independently calculated weight. The main difference lies in the fact that each neuron of the output layer is connected with all objects from the input layer, and their number is much higher than in neural networks used for predictive purposes. Cases positioned on the grid are not connected, although the cases that are in one given neuron are similar to those in a neighboring neuron.

The researcher can determine the size of the topological map, which refers to the probable maximum number of clusters. At this stage, one can use a priori knowledge gained while applying other clustering methods. In general, it is recommended to build large topological maps assuming that in each neuron there will be large enough number of cases.

4.4 Classification and regression trees

CART, which was developed by Breiman et al [31], is a recursive partitioning algorithm. It is used to build a classification tree if the dependent variable is nominal, and a regression tree if the dependent variable is continuous. The goal of this experiment is to predict the customers' response, which means that a classification model will be developed. To describe it briefly, a graphical model of a tree can be presented as a set of rules in the form of if-then statements. A visualization of a model is a significant advantage of that analytical approach from the marketing point of view. Prediction is an important task for marketing managers, but the knowledge of the interest area is crucial. Despite the fact that CART was introduced almost thirty years ago it has some important features, i.e. a priori probabilities and misclassification costs, which make it potentially useful in cost sensitive-learning.

4.5 Comparison and evaluation of models

To compare all models presented in that article the following metrics were used:

- Recall = TP / (TP + FN)
- Precision = TP / (TP + FP)
- Profit (see details in Table II).

The authors omitted the accuracy and true negative rate (Acc-) because the goal of the analysis is to predict object membership in the positive category. Acronyms used in the above formulas are derived from a known cost matrix, which is shown in Table I.

TABLE I
EXAMPLE OF COST MATRIX FOR TWO CATEGORIES OF DEPENDENT VARIABLES

		Classified	
		True	***False***
Observed	***True***	**TP** true positive	**FN** false negative
	False	**FP** false positive	**TN** true negative

For example, TP is an acronym for true positive, which means that an object belonging to the positive category was classified as positive. Higher costs are assigned to FP rather than to FN since researchers usually focus on predicting the positive class.

TABLE II
REVENUE-COST TABLE

	Revenue	Cost	Profit
TP	100	0.1	99.9
TN	0	-0.1	0.1
FP	0	0.1	-0.1
FN	-100	-0.1	-99.9

Additionally, the authors calculated the lift measures for the first few deciles of the test sample. A lift measure is the ratio between the modeled response and the random response. The modeled response is provided by a statistical or data mining tool and the predictive model is presented as a lift curve. The random response is sometimes called the base rate, and it represents the response percentage in the whole population. The lift measure used by the authors does not refer to the well known measure in association rules mining introduced by Brin et al [32].

5 Empirical evaluation

5.1 K-means clustering

When creating clusters the following variables were used: variables relating to the online activity of internet users (number of logins per month, number of logins within 6 months, all the days of unique logins, number of posts on forums, number of threads on forums, etc.), variables related to spending on services offered by the portal and games played by internet users. After standardization of 46 quantitative variables the principal components analysis (PCA) was conducted. On the basis of the Kaiser's criterion (eigenvalue > 1) 15 principal components were selected, which explained 75% of total variance. Then a representative variable with the highest factor loadings was selected from each of them.

As a result of the application of K-means algorithm, three clusters were built. As previously mentioned, the percentage of internet users who clicked on the banner amounted to 0.25% in the entire dataset. The percentage of response category in these clusters was: 0.25% (cluster 1), 0.36% (cluster 2), and 0.21% (cluster 3). In the next stage a one-way ANOVA was conducted, which enabled to select variables that best differentiate clusters (Table III). The number of the selected variables was limited to those for which the Sheffe post hoc test indicated the presence of differences between all clusters.

TABLE III
VARIABLES BEST DIFFERENTIATING CLUSTERS BUILT BY USING K-MEANS ALGORITHM

	cluster description summary				
K-means	all unique log days during the period considered	average daily number of logins in the last month	average spending on text messages (standardized value)	average spending on gifts for friends (standardized value)	Number of cases
cluster 1	411	1.38	-0.017	-0.004	25,664
cluster 2	177	1.05	-0.117	-0.057	19,003
cluster 3	610	2.64	0.229	0.074	36,917

5.2 SOM clustering

In the second approach, a set of 15 variables selected by PCA was also used. To ensure the comparability of results and a relatively large number of cases in each cluster the initial grid size of 2 x 2 was determined. A small size of one of the output neurons caused us to join two neighboring clusters and consequently we received three clusters. The percentage of positive categories in the three clusters obtained with the application of the Kohonen networks was as follows: 0.34% (cluster 1), 0.20% (cluster 2), and 0.24% (cluster 3). A brief description of the clusters is shown in Table IV. The one-way ANOVA and the Sheffe post hoc test were conducted here as well. It is worth noting that in the table there can be found the same variables that were present in the K-means method.

TABLE IV
VARIABLES BEST DIFFERENTIATING CLUSTERS BUILT BY USING KOHONEN NETWORKS

cluster description summary					
SOM	all unique log days during the period considered	average daily number of logins in the last month	average spending on text messages (standardized value)	average spending on gifts for friends (standardized value)	Number of cases
cluster 1	203	1.09	-0.108	-0.053	23,383
cluster 2	619	2.74	0.245	0.079	33,763
cluster 3	442	1.44	0.001	0.006	24,438

When looking at the content of clusters (Table V) it can be seen that 83% of cases belonging to cluster 1 (K-means) are in cluster 2 (SOM), 100% of cases from cluster 2 (K-means) are in cluster 1 (SOM), and 91% of cases from cluster 3 (K-means) are in cluster 2 (SOM).

TABLE V
COMPARISON OF CLUSTER'S CONTENT

		SOM			
		cluster 1	cluster 2	cluster 3	Total
K-means	cluster 1	4,380	0	21,284	25,664
	cluster 2	19,003	0	0	19,003
	cluster 3	0	33,763	3,154	36,917
	Total	23,383	33,763	24,438	81,584

It is acknowledged that the final number of clusters depends on their practical application. So it is in this case. The selection of the better method will depend on whether their combination with C&RT algorithm will provide a better predictive accuracy.

5.3 Predictive models

When building the predictive model we used a different set of independent variables. These included the demographic characteristics of users (sex, age, education) as well as variables relating to interactions with other users of the portal (number of friends, informing friends about birthdays, etc.).

Table VI illustrates information about the structure and size of learning samples and test samples. As previously mentioned symbols M1 and M3 refer to the hybrid models in which membership in the cluster is treated as an additional independent variable.

TABLE VI
STRUCTURE OF LEARNING SAMPLES

Model	Learning sample	Response percentage	Non-response percentage	Total learning sample	Test Sample
M0	L1 (unmodified)	104 (0.25%)	40,703 (99.75%)	40,807 (50.02%)	40,777 (49.98%)
	L2 (random under-sampling)	104 (10.00%)	936 (90.00%)	1,040	40,777
	L3 (two-sided sampling technique)	312 (10.00%)	2,808 (90.00%)	3,120	40,777
M1	L1 (unmodified)	116 (0.28%)	40,969 (99.72%)	41,085 (50.36%)	40,499 (49.64%)
	L2 (random under-sampling)	116 (10.00%)	1,044 (90.00%)	1,160	40,499
	L3 (two-sided sampling technique)	348 (10.00%)	3,132 (90.00%)	3,480	40,499
M2	L1 (unmodified)*	30 (0.23%)	12,823 (99.77%)	12,853 (50.08%)	12,811 (49.92%)
		30 (0.31%)	9,583 (99.69%)	9,613 (50.59%)	9,390 (49.41%)
		36 (0.19%)	18,582 (99.81%)	18,618 (50.43%)	18,299 (49.57%)
	L2 (random under-sampling)*	30 (10.00%)	270 (90.00%)	300	12,811
		30 (10.00%)	270 (90.00%)	300	9,390
		36 (10.00%)	324 (90.00%)	360	18,299
	L3 (two-sided sampling technique)*	90 (10.00%)	810 (90.00%)	900	12,811
		90 (10.00%)	810 (90.00%)	900	9,390
		108 (10.00%)	972 (10.00%)	1,080	18,299
M3	L1 (unmodified)	102 (0.25%)	40,714 (99.75%)	40,816 (50.03%)	40,768 (49.97%)
	L2 (random under-sampling)	102 (10.00%)	918 (90.00%)	1,020	40,768
	L3 (two-sided sampling technique)	306 (10.00%)	2,754 (90.00%)	3,060	40,768
M4	L1 (unmodified)*	36 (0.31%)	11,666 (99.69%)	11,702 (50.04%)	11,681 (49.96%)
		24 (0.14%)	16,871 (99.86%)	16,895 (50.04%)	16,868 (49.96%)
		27 (0.22%)	12,259 (99.78%)	12,286 (50.27%)	12,152 (49.73%)
	L2 (random under-sampling)*	36 (10.00%)	324 (90.00%)	360	11,681
		24 (10.00%)	216 (90.00%)	240	16,868
		27 (10.00%)	243 (90.00%)	270	12,152
	L3 (two-sided sampling technique)*	108 (10.00%)	972 (90.00%)	1,080	11,681
		72 (10.00%)	648 (90.00%)	720	16,868
		81 (10.00%)	729 (90.00%)	810	12,152

* The first line contains information about cluster 1, the second - cluster 2, and the third - cluster 3.

Symbols M2 and M4 represent the C&RT models built separately for each cluster. Symbols L1, L2, and L3 are related to the unmodified learning sample, random under sampling, and two-sided sampling technique respectively.

Hybrid models (M1-M4) were compared with the standard C&RT model (M0) which was based on the entire set of independent variables, i.e. demographic variables and variables relating to the interactions between users, as well as those that were used to build clusters. The procedure for creating a learning sample and a test sample was the same as in hybrid models. In the first approach, the learning sample was left unmodified (49.98% of total), and then under sampling and two-sided sampling techniques were used. With regard to misclassification costs, they were set at the level of 10:1 and 20:1. The authors treat them as relative penalty related to an incorrect classification.

Table VII compares the performance of different models according to monetary costs and benefits of an advertising campaign. Values highlighted with a shade of gray indicate a positive financial result. As can easily be noticed, the best results were achieved by hybridization of K-means and C&RT algorithm using two-sided sampling technique (L3). A positive financial result delivered by the standard C&RT model based on under-sampling (L2) can be somewhat surprising. In terms of monetary profits of a campaign hybrid models M2 and M4 (built for each cluster separately) did not come true.

TABLE VII
PERFORMANCE OF MODELS ACCORDING TO MONETARY PROFITS OF CAMPAIGN

Model	L1 costs 10:1	L1 costs 20:1	L2 costs 10:1	L2 costs 20:1	L3 costs 10:1	L3 costs 20:1
M0	-6,223.40	-5,883.20	-1,722.20	726.40	-3003.80	-2,854.00
M1	-5,050.90	-5,050.90	2,085.70	954.70	2,375.90	3,305.10
M2a*	-2,019.00	-2,048.80	-63.60	-776.00	-920.80	-870.00
M2b*	-2,867.80	-2,887.00	-1,673.20	-1,332.80	-1,128.60	-468.40
M2c*	-2,170.50	-2,176.70	-755.50	93.70	-887.50	-332.90
M3	-6,424.30	-6,261.10	1,121.90	1,973.70	533.10	1,096.30
M4a*	-3,232.50	-3,232.50	-880.70	-932.90	-455.50	-766.70
M4b*	-2,713.60	-2,713.60	-1,771.40	-1,771.40	-1,567.80	-1,702.00
M4c*	-1,985.00	-2,002.80	-262.60	-430.40	225.60	-1,003.60

* Letter symbols a) to c) refer to clusters 1-3

Tables VIII and IX display performance of models according to the recall and the precision. To compare the differences between the models the G-test at the 95% confidence interval was conducted [33]. The best recall is provided by model M1 (combination of K-means with C&RT) based on L3 (two-sided sampling technique) with misclassification costs of 20:1. As to the precision, it is hard to decide clearly which model and sampling method is superior. Models marked with "xxx" classified all instances as non-response.

TABLE VIII
PERFORMANCE OF MODELS ACCORDING TO RECALL

Model	L1 costs 10:1	L1 costs 20:1	L2 costs 10:1	L2 costs 20:1	L3 costs 10:1	L3 costs 20:1
M0	0.000	0.019	0.350	0.524	0.233	0.252
M1	0.000	0.000	0.637	0.582	0.659	0.747
M2a*	0.000	0.000	0.424	0.303	0.273	0.273
M2b*	0.000	0.000	0.211	0.263	0.289	0.395
M2c*	0.000	0.000	0.250	0.525	0.250	0.400
M3	0.000	0.010	0.590	0.686	0.505	0.562
M4a*	0.000	0.000	0.341	0.341	0.409	0.364
M4b*	0.000	0.000	0.205	0.205	0.227	0.205
M4c*	0.000	0.000	0.406	0.406	0.500	0.219

* Letter symbols a) to c) refer to clusters 1-3

TABLE IX
PERFORMANCE OF MODELS ACCORDING TO PRECISION

Model	L1 costs 10:1	L1 costs 20:1	L2 costs 10:1	L2 costs 20:1	L3 costs 10:1	L3 costs 20:1
M0	xxx	0.007	0.003	0.003	0.003	0.003
M1	xxx	xxx	0.003	0.002	0.003	0.003
M2a*	xxx	0	0.003	0.003	0.003	0.003
M2b*	0	0	0.004	0.004	0.005	0.005
M2c*	xxx	0	0.003	0.002	0.003	0.002
M3	xxx	0.005	0.003	0.002	0.003	0.003
M4a*	xxx	xxx	0.005	0.004	0.004	0.004
M4b*	xxx	xxx	0.002	0.002	0.002	0.002
M4c*	xxx	0	0.003	0.002	0.003	0.003

* Letter symbols a) to c) refer to clusters 1-3

TABLE X
PERFORMANCE OF MODELS ACCORDING TO CUMULATIVE LIFT MEASURES FOR 1ST AND 2ND DECILES

	Model	L1 costs 10:1	L1 costs 20:1	L2 costs 10:1	L2 costs 20:1	L3 costs 10:1	L3 costs 20:1
1st decile	M0	1.44	1.40	0.82	1.08	1.39	1.25
	M1	1.34	1.34	1.12	1.45	1.49	1.20
	M3	1.84	1.50	1.11	1.04	0.94	1.30
2nd decile	M0	1.41	1.40	0.82	1.08	1.39	1.25
	M1	1.34	1.34	1.35	1.35	1.18	1.20
	M3	1.48	1.33	1.11	1.04	1.13	1.15

As for the lift measures values, they are shown in Table X. Shaded areas in the table refer to lift measures higher than 1.4. We limited the calculation only to models based on the entire set of observations (M0, M1, and M3) to ensure their comparability. The best results were highlighted gray. If a

company intended to reduce spending on a promotional campaign and displayed a banner ad to 10 percent of current users, model M3 (SOM-C&RT) would be the best solution. For 20% of users the highest lift measure was again obtained from model M3. It is worth noting that both hybrid models were based on the unmodified learning sample.

6 Conclusions

The conducted analyses show that the best results were obtained by combining K-means algorithm with C&RT algorithm, where information about belonging to clusters is treated as an additional independent variable in the model. Model M1 outperformed other models in terms of the profit and the recall. It is worth noting that one of these additional variables was involved in the partition of the tree, although its position in the final predictor variables ranking was relatively low.

Unfortunately, building separate C&RT models for particular clusters did not meet the authors' expectations. The results obtained in this manner were even worse than the results provided by the standard C&RT model with the whole set of independent variables. It seems that with such a highly skewed distribution of dependent variables one should use more sophisticated methods of overcoming this problem, or rely on ensemble models thus giving up merits of the content-related interpretation of the model.

7 References

[1] A. Goldfarb A., C. Tucker, "Online Display Advertising: Targeting and Intrusiveness", in *Marketing Science*, Vol. 30 No. 3, May-June 2011, pp. 389-404.

[2] N. Hollis, "Ten years of learning on how online advertising builds brands", in *J. Advertising Res.* 45(2), 2005, pp. 255–268.

[3] J. Surma, A. Furmanek, "Data mining in on-line social network for marketing response analysis", in *The Third IEEE International Conference on Social Computing (SocialCom2011)*, MIT, Cambridge, 2011.

[4] M. Richardson, E. Dominowska, and R. Ragno, "Predicting Clicks: Estimating the Click-Through Rate for New Ads", in *Proceedings of the Sixteenth International World Wide Web Conference*, Banff, Canada, 2007, pp. 1-9.

[5] Ch-J. Wang, H-H. Chen, "Learning user behaviors for advertisements click prediction", *in Proceedings of the 34rd international ACM SIGIR conference on research and development in information retrieval Workshop on Internet Advertising*, Bejing, China 2011, pp. 1-6.

[6] W-S. Yang, J-B. Dia, H-C. Heng, and H-T. Lin, "Mining Social Networks for Targeted Advertising", in *Proceedings of the 39th Hawaii International Conference on System Sciences*, 2006, pp. 1-10.

[7] F. Provost, B. Dalessandro, R. Hook, X. Zhang, and A. Murray, "Audience Selection for On-line Brand Advertising: Privacy-friendly Social Network Targeting", KDD'09, June 28–July 1, 2009, Paris, France, pp. 1-9.

[8] M. Ben-Akiva et al., "Hybrid choice models: progress and challenges", in *Marketing Letters* 13:3, 2002, pp. 163–175.

[9] L. Breiman, "Random forests," in *Machine Learning*, 45, Kluwer Academic Publishers, 2001, pp. 5-32.

[10] J. H. Friedman, *Greedy function approximation: a gradient boosting machine*, Technical Report, Department of Statistics, Stanford University, 1999.

[11] J. H. Friedman, *Stochastic gradient boosting*, Technical Report, Department of Statistics, Stanford University, 1999.

[12] H. Wang, Y. Yu, and K. Zhang, "Ensemble probit models to predict cross selling of home loans for credit card customers", in *International Journal of Data Warehousing and Mining*, Volume 4, Issue 2, 2008, pp. 15-21.

[13] M. Wei, L. Chai, R. Wei, and W. Huo, "A solution to the cross-selling problem of PAKDD-2007: Ensemble model of treeNet and logistic regression", in *International Journal of Data Warehousing and Mining*, Volume 4, Issue 2, 2008, pp. 9-14.

[14] I. Bose and X. Chen, "Hybrid models using unsupervised clustering for prediction of customer churn", in *Journal of Organizational Computing and Electronic Commerce*, Vol. 19, No. 2, April-June 2009, pp. 133-151.

[15] Y. Li, Z. Deng, Q. Qian, and R. Xu, "Churn forecast based on two-step classification in security industry", in *Intelligent Information Management*, 2011, 3, 160-165.

[16] A. K. Kirshners, S. V. Parshutin, and A. N. Borisov, "Combining clustering and a decision tree classifier in a forecasting task" in *Automatic Control and Computer Sciences*, 2010, Vol. 44, No. 3, pp. 124–132.

[17] D. Qiu, Y. Wang, and B. Bi, "Identify cross-selling opportunities via hybrid classifier", in *International Journal of Data Warehousing and Mining*, Volume 4, Issue 2, 2008, pp. 55-62.

[18] A. Martin, V. Gayhatri, G. Saranya, P. Gayhatri, and P. Venkatesan, "A hybrid model for bankruptcy prediction using genetic algorithm, fuzzy c-means and MARS", in *International Journal on Soft Computing (IJSC)*, Vol.2, No.1, February 2011, pp. 12-24.

[19] A. Brabazon and P. B. Keenan, "A hybrid genetic model for the prediction of corporate failure", in *Computational Management Science*, Springer Verlag, 2004, pp. 293-310.

[20] W. E. Lindahl and C. Winship, "A logit model with interactions for predicting major gift donors, in *Research in Higher Education*, Vol. 35, No. 6/1994, pp. 729-743.

[21] D. Steinberg and N. S. Cardell, "The hybrid CART-logit model in classification and data mining", 1998 [Online], Available: http://www.salford-systems.com.

[22] C. Chen, A. Liaw, and L. Breiman, "Using random forest to learn unbalanced data," in *Technical Report*, 666, Statistics Department, University of California, Berkeley, 2004.

[23] D. A. Cieslak and N. V. Chawla, "Learning decision trees for unbalanced data," in *ECML/PKDD*, 2008.

[24] J. Surma, M. Łapczyński, "Selecting data mining model for web advertising in virtual communities", in *Proc. The First International Conference on Advances in Information Mining and Management (IMMM 2011)*, Barcelona, Spain, 2011, pp. 107-112.

[25] C. X. Ling and V. S. Sheng, "Cost-Sensitive Learning and the Class Imbalance Problem", in *Encyclopedia of Machine Learning*, C. Sammut, Ed. Springer Verlag, Berlin, 2008.

[26] R. C. Blattberg, B-D. Kim, S. A. Neslin, *Database marketing. Analyzing and managing customers*, Springer, New York 2008.

[27] P. Kline, *An easy guide to factor analysis*, Routledge, New York 1994.

[28] M. J. A. Berry, G. S. Linoff, *Data mining techniques for marketing, sales, and customer relationship management. Second Edition*, Wiley Publishing Inc., Indianapolis, Indiana, 2004.

[29] G. Gan, Ch. Ma, and J. Wu, *Data clustering. Theory, algorithms, and applications*, ASA-SIAM Series on Statistics and Applied Probability, SIAM Philadelphia, ASA, Alexandria, VA, 2007.

[30] T. Kohonen, "The Self-Organizing Map", in *Proceedings of the IEEE*, 78:, 1990, pp. 1464-1480.

[31] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and regression trees*, Belmont, CA: Wadsworth International Group, 1984.

[32] S. Brin S., R. Motwani, J. D. Ullman, and S. Tsur, "Dynamic itemset counting and implication rules for market basket data", in Peckham, J., ed.: *Proceedings ACM SIGMOD International Conference on Management of Data*, May 13-15, 1997, pp. 255-264.

[33] R. R. Sokal, *Biometry: the principles and practice of statistics in biological research*, New York, Freeman, 1981.

Application of Data Mining Techniques to Predict Allergy Outbreaks among Elementary School Children

Integration of Hourly Air Pollution, Bi-Daily Upper-Air, and Daily School Health Surveillance Systems in Pennsylvania

Ahmed YoussefAgha, PhD
Dept. of Applied Health Science
Indiana University
Bloomington, Indiana, USA
E-mail: ahmyouss@indiana.edu

Wasantha Jayawardene, MD
Dept. of Applied Health Science
Indiana University
Bloomington, Indiana, USA
E-mail: wajayawa@indiana.edu

David Lohrmann, PhD
Dept. of Applied Health Science
Indiana University
Bloomington, Indiana, USA
E-mail: dlohrman@indiana.edu

Gamal El Afandi, PhD
College of Agricultural, Environmental & Natural Sciences
Tuskegee University
Tuskegee, Alabama, USA
E-mail: gelafandi@mytu.tuskegee.edu

Abstract: **Objectives of this study are to determine if a relationship exists between occurrence of allergies among elementary school children and daily upper-air observations (temperature, relative humidity, dew point, mixing ratio) and daily air pollution (particulate matter, sulfur dioxide, nitrogen dioxide, carbon monoxide, and ozone); and, if so, to derive a mathematical model that predicts allergies. Using an ecological study design, school health records of 168,825 students in elementary schools enrolled in "Health eTools for Schools" within 49 Pennsylvania counties were analyzed. Upper-air measurements from ground level to the 850mb pressure level and air pollution measurements were obtained. Appropriate data mining techniques were utilized to validate and integrate three databases. Binary logistic regression used for analysis. A Generalized Estimating Equation model was used to predict the occurrence of more than 13 cases, the daily mean for 2008-2010. Results showed that the prevalence of allergies among school children in Pennsylvania increased over last three years. The primary occurrence of allergies among school children was in August-September, followed by December and April, while the lowest in January and May. Upper-air temperature and mixing ratio, as well as SO_2, CO, O_3, PM_{10} were significantly associated with occurrence of allergies ($p<0.01$). In conclusion, monitoring of upper-air observation and air pollution data over time can be a reliable means for predicting outbreaks of allergies among elementary school children. Such predictions could help parents and school nurses implement effective precautionary measures.**

Keywords: air pollutants, upper-air indicators, allergies, school health records, Pennsylvania

I. INTRODUCTION

Hay fever, respiratory allergies, food allergies, and skin allergies are the main types of allergies among children in the US. Percent of children with diagnosed hay fever in the past 12 months is 9.5% (7.1 million), while 11.5% (3.4 million) of children reported as having respiratory allergies. In addition, 4.6% (9.4 million) of children had food allergies in the past 12 months, while 12.6% of children reported were diagnosed with skin allergies [1]. Seasonal allergies are fairly common in children older than age five. According to the American Academy of Allergy, Asthma, and Immunology, about 10-15% of school-age children have seasonal allergies [2].

Seasonality of allergies [3, 4] and their association with asthma [5] are well-known phenomena. Atopic eczema and allergic rhinitis were found to be higher in period September–May. Some evidence support the assertion that non-summer warmth and urban air pollution [6], probably mediated through exposure to common allergens such as dust mites, are possible risk factors for allergies in school-aged children [7]. Meteorological factors greatly influence the occurrence of pollen grains in the air. Dry and hot weather speeds up maturation and the loosening of pollen grains from anthers, and the concentration of pollen grains is considerably higher than in cold and wet weather [8]. Temperature has a positive correlation with pollen count, while relative humidity has an inverse correlation with the pollen count in the atmosphere. Reportedly, the occurrence of allergic diseases is usually associated with increase in the level of CO (carbon monoxide), SO_2 (sulfur dioxide), PM_{10} and $PM_{2.5}$ (particulate

matter with a diameter of <10μm and <2.5μm, respectively), and O_3 (ozone), after adjusting for confounders [6].

The current study was undertaken to determine if a relationship exists between occurrence of allergies among school children, based on school health services records, and routine upper-air variables, such as mixing ratio, relative humidity, temperature, and dew point, as well as six air pollutants mentioned above. Although temperature, relative humidity, and air pollutants at ground level have been considered in allergy studies, neither upper-air dew point nor mixing ratio have been linked with allergy or used for prediction of outbreaks. As the amount of humidity is relative to the temperature available to do the work of evaporation, and can fluctuate even if no change occurs in the actual amount of water vapor per unit mass of dry air (mixing ratio), dew point and mixing ratio provide the best assessment of humidity as it relates to some allergies.

School health records have been utilized for allergy-tracking because they are readily accessible through the existing school data collection and storage infrastructure [9]. In many Pennsylvania schools student health records are compiled through a web-based software application portal called "Health eTools for Schools"(hereafter referred to as "eTools") that was funded by the Highmark Foundation through its Healthy High Five Initiative [10] and developed by InnerLink, Inc., a private, for-profit company. With the utilitarian purpose of providing data for state and local health planning, eTools is offered free of charge to local public, private and parochial schools. It allows nurses to efficiently enter and download student health data, including records of daily clinic visits for allergic conditions, using a hand-held electronic device, and via computerized programming, compile and submit required annual reports to the Pennsylvania State Health Department.

Public health is an information-intensive field, which needs timely, accurate, and authoritative information from a variety of sources [11-13]. According to "The Future of Public Health" report of Institute of Medicine in 1988, which launched a series of public health reforms that continues to this day, the essence of community health assessment is the collection, analysis, interpretation, and communication of data and information from various sources [14].

Consistent with the literature, the current study was designed to investigate the possible relationship between allergic diseases among school children on a particular day and the effect of air pollution indicators, i.e., PM_{10}, $PM_{2.5}$, SO_2, NO_2, CO, and O_3, throughout the day by using data mining techniques. The ultimate purpose was to determine if a mathematical model could be derived to predict daily allergy burden based on the combination of upper air data plus air pollution levels and student health record entries from Pennsylvania schools.

II. METHODS

A. Study Design

Ecological study design was adopted in order to understand the relationship between daily occurrence of allergies in a "whole population" of elementary school students and daily measurements of upper-air observation parameters [15]. In the ecological design, which investigates group-level variables, a geographical region can be analyzed in a cross-sectional manner (once or repetitively) to investigate the variation in a health-related variable (e.g., mean blood pressure, hospitalizations for allergy, and homicide rates) and its associations with regional characteristics (e.g., salt intake, air pollution, handgun laws, and drug policies) [15].

An ecological design has the advantageous ability to control for individual-level variability while at the same time addressing influences at the regional-level. In addition, this study design enables researchers to include all the students of the enrolled elementary schools in the study sample, contrary to one in which each student with an allergic condition serves as his/her own control and excludes other students [15]. Moreover, extracting daily upper-air data from existing environmental databases and retrieving computerized daily health data from existing repositories, such as eTools, is inherently cost-effective in that it requires a very low level of effort.

B. Study Population

School districts located in 49 of the 67 counties in Pennsylvania participated in eTools. From 2008 to 2010, enrolled school districts received eTools services for 168,825 elementary school students. These students constituted the study population. All eTools services for participating school districts were subsidized by the funders of eTools (Highmark Foundation) and covered costs of utilizing eTools through a three-year period. At the individual level, the sole participation eligibility requirement was to be a student with a health record in an elementary school that utilized eTools. Student records from these schools were excluded if they contained incomplete or inaccurately entered health record data. In addition, no race-, ethnicity-, or income-based bias existed in the enrollment of schools districts, schools, or students in eTools. All of the above factors are important in respect to generalizability of study results to elementary school students at the state level or beyond the state level [15].

C. Data Collection

Within the 49 Pennsylvania counties, 168,825 records of elementary school students were identified for this study. Data on allergies were originally noted in records maintained by school nurses as the type of treatment given to a student on a particular day. Treatment options for school health nurses were based on the prescribed medication provided for the student. For purposes of this study, having allergy was defined as "any case managed with antihistamines". In fact, school nurses recorded the trade name of all the medications administered,

which were then later categorized into functional groups (i.e., antihistamines, analgesics, etc.). It was assumed that the treatment option noted by a school nurse correctly represented the disease (figure-1).

Data analyzed for this study were provided to the authors by InnerLink, Inc. which had custodial responsibility under a common Statement of Understanding and Service Level Agreement with all participating school systems (table-1). The school year in Pennsylvania usually begins in August, ends in June and typically includes an extended winter break. Therefore, surveillance data were commonly unavailable for the last three weeks of June, all of July, the first three weeks of August, and the second half of December every year. According to Pennsylvania State regulations, the healthcare provider should state whether the child is qualified and able to self-administer the medication. The number of visits by middle school and high school students to school health nurse's office, therefore, may not accurately represent the number of allergy occurrences. Hence, only data from elementary school students' records were included in this study (figure-1).

The upper-air data used in this study were downloaded from the Wyoming University web site [16]. Measurements were obtained from the ground level to 1500 meters level (equivalent to 850 mb pressure). These upper-air observation values cover an area with a radius of about 800 kilometers, around Pittsburg, PA. As this coverage area includes the entire state of Pennsylvania, upper-air measurements, unlike ground level air-pollution, were consistent for all eTools schools regardless of geographical location. Upper-air observation values are obtained twice every day: at 0Z and at 12Z. The Z-time is the basis for synoptic meteorology, which requires collection of all measurements at the same time every day and, thereby, produces a snapshot of the state of the atmosphere worldwide. The 0Z time in the US is 6 p.m. in Eastern Standard Time (EST) and 7 p.m. in Eastern Daylight Time (EDT), while 12Z time in the US is 6 a.m. in EST and 7 a.m. in EDT. For each day of the three years (2008-2010) the measured values at time 12Z were obtained for analysis. As the 0Z (6 p.m. EST and 7 p.m. EDT) is irrelevant to asthma exacerbations that occur during elementary school hours, authors had to choose the 12Z time for the analysis (figure-1).

Hourly data on six air pollutants were extracted from EPA measurement stations throughout the state. Therefore, unlike in upper-air variables, the geographical distribution of schools within the state affected the study results derived from air-pollution variables (Figure-1). These were obtained for school days only, although the data are available for all days throughout the period 2008-2010. Data for 1am–3pm period were obtained, because it represents the hours before and during school time. After school hours were ignored as the pollutant levels during these hours do not contribute to allergies that occur at school (figure-1).

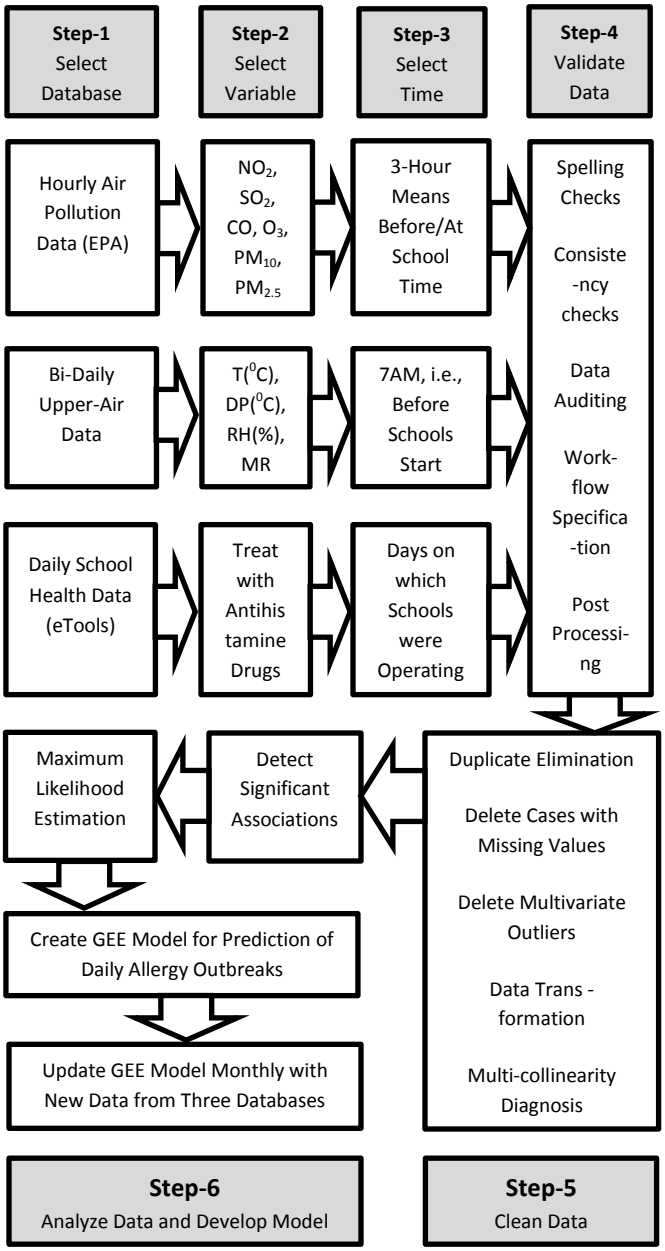


Figure 1: Data Mining Algorithm to Create a GEE Model for Prediction of Allergy Outbreaks in School Children
[T=upper-air temperature; DP=upper-air dew point; RH=upper-air relative humidity; MR=upper-air mixing ratio; EPA=Environmental Protection Agency; GEE=Generalized Estimation Equation; NO2=nitrogen dioxide; SO2=sulfur dioxide; CO=carbon monoxide; O3=ozone; PM10 and PM2.5=particulate matter with <10 and <2.5 μm respectively]

D. Analysis

We performed the allowed character checks, cardinality check, consistency checks, data type checks, limit checks, logic check, spelling and grammar check in the process of data validation (figure-1). In addition, missing values were deleted, duplicate values were eliminated, and multivariate outliers were deleted. Data transformation was performed when

required. Multi-collinearity diagnostics were also performed with SAS.

To minimize the effect of the increase in asthma reporting over three years, the ratio of daily and weekly allergy cases to the annual average rather than the absolute number of daily allergy cases was utilized for analysis. Similarly, the ratio of weekly mean of asthma exacerbations to the annual average was used instead of the absolute number of asthma exacerbations per week in order to minimize the effect of the distortion (as indicated above) in the weekly pattern of asthma exacerbation occurrences.

We used binary logistic regression modeling to analyze data from three large databases using SAS. Logistic regression is a useful method utilized widely in data mining applications [17]. It is more advanced than the chi-square test, which allows the analysis of only two categorical variables. It allows the researcher to use both continuous and categorical predictors in modeling. Binary logistic regression converts the binary response into a logit value (the natural logarithm of the odds of the event occurring or not) and then establishes maximum likelihood estimates (MLE). The assumptions of this analysis are particularly applicable to the current study, as we used large databases with large sample asymptotic property. This means that the reliability of the estimates is high when a large number of cases for each observed combination of X is available [17]. In addition, in binary logistic regression, changes in the response itself are not modeled; instead changes in the log odds of the response are modeled. Another important feature is that binary logistic regression assumes neither the linearity of the relationship between predictor variables, nor homoscedasticity and normality of residuals. Model appropriateness was confirmed with the Wald statistic that tested the significance of individual parameters [17].

Generalized Estimating Equation (GEE) model was used to predict the probability of occurrence of cases greater than or equal to 13 (a cutoff value equal to the 2008-2010 mean). The repeated subject (cluster) in the model is the "matched date," for example, March 8th in 2007, 2008, 2009, and in 2010 is a subject. GEE needs at least 100 clusters to study 5-12 exploratory variables, however, a great deal of confidence is assured with 200 clusters, [18, 19] the circumstance in in the current study. "Logit" link function on binary distribution was used on a binary dependent variable. "P" represents DailyAllergyCases≥13 and "1-P" represents DailyAllergyCases<13. The independent variables were the upper-air observation variables.

To avoid the complexity of obtained GEE model, we adhered to a model with a binary dependent variable (DailyAllergyCases<13 and DailyAllergyCases≥13), rather than having three levels (low, average, and high) of occurring daily allergy cases. In our future study, which will also include pollen data, we will develop separate models for pollen season and non-pollen season. In the future study, we will also incorporate machine learning approaches, in addition to statistical methods.

Table-1: Demographic Characteristics of Students

Variable	Student Category	Percentage
Gender	Males	50.80%
	Females	49.20%
Race	White Alone	74.78%
	African-American Alone	8.30%
	Hispanic Alone	7.91%
	Other	3.10%
	Multi-Race	4.27%
Free-Lunch Eligibility	Eligible (Low-Income)	38.36%
	Ineligible	61.64%
Urban-Rural Distribution	Rural	39.22%
	Suburban	42.65%
	Urban	18.13%

III. RESULTS

Demographic characteristics of 168,825 elementary school students are shown in table-1 [20, 21]. Offices of school health nurses in Pennsylvania had 259,951 visits of elementary school students for various health problems during the three year period, 2008-2010. A gradual increase of allergy cases was observed from 2008 to 2010, both in elementary schools and in schools with higher grades. This is partially explained by the increase in the number of schools participating in the allergy surveillance system. The total number of schools (elementary, middle, and high) increased from 2008 to 2009, with a slight decrease from 2009 to 2010. However, a boost from 0.85 to 1.77 (208% increase) in the ratio of daily average of allergy cases to number of schools (preK-5) partially explained the real increase in allergy prevalence among school children, because that particular increase is not influenced by an increase in the number of schools participating in the allergy surveillance system. It may, however, be influenced by increased reporting of school nurses, since any surveillance system takes a few years to reach the full functional capacity. Geographical differences in new school enrolment may also have influenced this 208% increase.

Use of antihistamines among school children as reported by school health nurses has an almost identical pattern in each year during 2008-2010. Peak of antihistamine use lies in period from late August to early September, although there are noticeable increases in December (early winter) and in April (early spring) too. The lowest use of antihistamine is reported in January and in May. The summer season compared to winter, had 17 times higher tendency of exceeding daily mean of allergy cases (7 cases), which was followed by fall (2.6 times higher), and spring (2.1 times higher). In other words, 68% of days in summer exceeded daily mean of allergy cases for three years, followed by fall (38%), and spring (35%).

Upper-air temperature peaks in July (hot) and has its minimum in January (cold). There were significant differences between temperature means for each season ($p<0.001$), with the highest in summer, followed by spring, and then fall. Similarly, upper-air dew point peaks in July (wet) and reaches

its minimum in January (dry). Significant differences existed between seasons ($p<0.001$), with the highest in summer, followed by spring, and then fall. Significant differences between mixing ratio means for each season were also statistically significant ($p<0.001$), with the highest in summer, followed by spring, and fall. Variation in upper-air relative humidity is complex and it's difficult to identify a specific pattern. Differences between relative humidity means for each season were statistically significant ($p<0.02$), with the highest in summer, followed by winter, and then fall. The lowest relative humidity was recorded in spring. It was revealed that upper-air temperature and mixing ratio were important in determining whether or not the number of allergy cases is greater than the daily mean of allergy cases for the last three year period (>13 cases).

Peak hours of NO_2 were 4–6 am, followed by the remaining morning hours (1–3 am, 7–9 am, and 10–12 noon). Conversely, SO_2 levels were highest towards noon (10–12 noon), followed by 7–9 am and 1–3 pm with the lowest levels recorded in the early morning (1–6 am). Highest levels of CO were recorded for the morning hours when students leave for school (7–9 am), followed by earlier hours in the morning (1–6 am). As with NO_2, the CO levels decreased towards noon and further decreased during the afternoon. The peak for $PM_{2.5}$ occurred at 7–9 am, while the afternoon hours had the lowest. PM_{10} levels did not vary significantly compared to other pollutants and levels gradually decreased from 4 am to 3 pm. PM_{10} data for 1–3 am and 1-3 pm time intervals were not available.

Table 2: Modeling the Probability that the Number of Allergy Occurrences ≥ 13 among preK-5 Students on a Particular Day Based on Pollutant and Upper-Air Parameters: Analysis of GEE Parameter and Empirical Standard Error Estimates

Parameter	**Estimate**	**Error**	**Z**	**Prob. >|Z|**
Intercept	-2.337	0.256	-9.1	<.0001
T	0.087	0.031	2.8	0.0054
T *$O3_{DAY}$*$CO_{1\text{-}3AM}$	0.018	0.004	4.7	<0.0001
MR*$PM10_{7\text{-}9AM}$ *$SO2_{1\text{-}3AM}$	-0.003	0.001	-3.9	<0.0001

$O3_{DAY}$ = mean O3 concentration of the previous day
$SO2_{1\text{-}3AM}$ = mean SO2 concentration for the 1 am – 3 am time period of the previous day
$CO_{1\text{-}3AM}$ = mean CO concentration for the 1 am – 3 am time period of the previous day
$PM10_{7\text{-}9AM}$ = mean PM_{10} concentration for the 7 am – 9 am time period of the previous day
MR = Upper-air mixing ratio (in g/kg) of the previous day
T = Upper-air temperature (in Celsius) of the previous day

SO_2, CO, O_3, and PM_{10} were significant in determining whether or not the number of allergy cases is greater than the daily mean of allergy cases for the last three year period (>13 cases). However, none of these air pollutants predicted allergy occurrences in their univariate relationships with allergy cases. In other words, the interaction of $O3_{DAY}$ and $CO_{1\text{-}3AM}$ with temperature was significant, while the interaction effect of $PM10_{7\text{-}9AM}$ and $SO2_{1\text{-}3AM}$ with mixing ratio was significant in predicting allergy cases. Therefore, the results indicate the importance of integrating upper-air and air pollution databases.

Using the GEE model above (table-2), the following equation was adopted for determining the probability odds ratio (y) of having ≥13 (daily-mean for a period of three years + one standard deviation) on the next day:

$$y = a_0 + a_1x_1 + a_2x_2 + a_3x_3 + a_4\,(x_4{*}x_2)$$

$$\text{Log}\frac{P}{1-P} = \begin{array}{l} -2.3372 + 0.0874\,(T) \\ +\,0.0180\,(T * O3_{DAY} * CO_{1\text{-}3AM}) \\ -\,0.0031\,(MR * PM10_{7\text{-}9AM} * SO2_{1\text{-}3AM}) \end{array}$$

log P/(1-P) = Logarithm of odds ratio for the [mean + one standard deviation] cases (=13 cases) on a day
P = Probability of having ≥13 allergy cases on a particular day
1 – P = Probability of having <13 allergy cases on that day
T = Upper-air temperature ^{0}C (in Celsius)
MR = Upper-air actual mixing ratio g/kg (in grams per kilogram)
$O3_{DAY}$ = mean O3 concentration
$CO_{1\text{-}3AM}$ = mean CO concentration for the 1 am – 3 am time period
$PM10_{7\text{-}9AM}$ = mean PM_{10} concentration for the 7 am – 9 am time period
$SO2_{1\text{-}3AM}$ = mean SO2 concentration for the 1 am – 3 am time period

As an example, if the upper-air temperature and upper- air mixing ratio were 9^0C and 3 g/kg respectively and the levels of O_3, SO_2, CO, and PM_{10} were 19.54, 5.34, 0.28, and 20.55 respectively on a particular day, the probability (P) of having more than 13 allergy cases within the surveillance system on the next day is 68%.

IV. DISCUSSION

The surveillance of allergy among school children in Pennsylvania has improved during the last four years in both aspects: the school enrollment in the surveillance system and the disease reporting by school health nurses, a proven, reliable data source [22]. The increased exposure to allergens and changes in other socioeconomic and sociodemographic factors among school children may be the main reasons for increases in allergies in Pennsylvania. The finding of the current study that peaks (main outbreaks) of allergy occurrence were reported in August-September and in April was compatible with evidence from other empirical studies. However, having the highest prevalence in summer months

may be associated with characteristics of pollen that cause allergic symptoms [2].

Annual averages of upper-air observations (temperature, relative humidity, dew point, mixing ratio) did not differ between years. As explained previously, a higher temperature (greater than three year median) is associated with occurrence of more allergies among Pennsylvania school children. However, this study considered only the upper-air temperature, not the surface level ambient temperature. In addition, a higher upper-air mixing ratio was associated with occurrence of more allergies. As discussed previously, the concept of relative humidity is complex, because relative humidity is relative to the amount of energy (measured by temperature) available to do the work of evaporation, and it can be changed without having any change in the actual amount of water vapor in the air (absolute humidity). Therefore, the actual mixing ratio in conjunction with temperature provides a better assessment of humidity, which is more relevant to allergies. The interaction of $O3_{DAY}$ and $CO_{1\text{-}3AM}$ with temperature was significant, while the interaction effect of $PM10_{7\text{-}9AM}$ and $SO2_{1\text{-}3AM}$ with mixing ratio was significant in predicting allergy cases. This encourages the integration of multiple databases in the process of prediction.

Put another way, the routinely measured and publicly available upper-air indicators provide a good estimate and a reliable tool for forecasting allergy burden on the school health system today or on a future day. This output of analysis with GEE is based on a three-year mean of allergies (13 cases) and does not restrict the utilization of the above equation beyond interpolation. Therefore, the extrapolation for forecasting is also realistic given the current analysis, based on the average of allergies for the 2008-2010 period. At last, this model should be updated monthly using new data that will be obtained from three databases.

However, there are several limitations in this study. There is a possibility that other allergy triggers, such as pollen and respiratory infections, may confound the relationship between upper-air variables and allergy occurrences. Allergy surveillance in schools is not being conducted during winter break and summer vacations, as well as during weekends and school holidays. The relatively less data available for the summer season due to school vacation, may somewhat distort the relationships caused by unequal representativeness across seasons.

In addition, it is difficult to exclude the possibility that some students who experience worsening of symptoms at night or early in the morning did not attend the school so their allergies are not included in school health records. It was also impossible to estimate the number of students with allergy symptoms who did not receive medications in school, because some students prefer taking medications before going to school or after school. In addition, there is also a possibility that some elementary school students may have taken allergy medications without informing the school nurse. Therefore, we cannot exclude the possibility that use of antihistamine medications as recorded by school nurse somewhat under-represented the occurrence of allergy symptoms at school. Unavailability of antihistamine medications usage data at school level as well as absence of data on demographic characteristics (e.g., ethnicity, income) of students in each school or school district were also limitations.

The number of schools using eTools changed each year with some schools dropping off and others joining, causing fluctuations in the total number of available student data records. Additionally, each year some students were leaving their school or school district. Nevertheless, the total number of student data strings provided for any one year was sufficiently robust, as was the number of data strings available for multi-year comparisons, to generate reliable results.

V. CONCLUSION

Monitoring of upper-air observation data and air pollution data over time using data mining techniques can serve as a reliable means for predicting increased occurrence of allergies among elementary school children. The new mathematical model derived from statistical integration of routine environmental observations and school health records may be used to scrutinize the complexity of allergic diseases as a dynamic outcome determined by multiple environmental parameters. It's also possible to assess the risk for future allergy outbreaks based on fluctuation analysis of a long time series of atmospheric function for taking more effective precautionary measures. Appropriate data mining techniques are proven to be useful tools in this process of prediction.

Predicting of allergy outbreaks is important for children diagnosed with allergies as their household members can take more preventive measures to avoid allergy occurrences, regardless of the disease severity. In addition, being informed of a possible allergy outbreak on a particular day or week, school nurses and teachers will be able to pay more attention on early identification of allergy occurrences among children in the classroom. Similarly, doctors and nurses in emergency room or in the family practice can take extra precautions and utilize resources more efficiently to serve an increased number of allergy patients.

With accurate prediction of increased number of allergy occurrences, the school health system can play a major role by effectively reallocating both human and physical resources as well as by alerting teachers and vulnerable children to take extra precautions that will manage allergy outbreaks very early. It also enables to identify and quickly treat those allergies that do occur. Mass media and local media can also play an important role in effort. Results of the current study can be used to inform similar programming at the national level and in other states.

ACKNOWLEDGEMENTS

Authors thank InnerLink Inc. for developing eTools and also Highmark Foundation of Pittsburgh, Pennsylvania for

implementing Highmark Healthy High 5 Initiative. The authors report no conflicts of interest in relation to the current study.

REFERENCES

[1] C. f. D. C. a. Prevention. (2011, Aug 21, 2011). *FastStats: Allergies and Hay Fever*. Available: http://www.cdc.gov/nchs/fastats/allergies.htm

[2] A. American Academy of Allergy, and Immunology. (2011, Aug 14, 2011). *Allergies.* Available: http://www.aaaai.org/conditions-and-treatments/allergies.aspx

[3] N. Ricard, L. Sauriol, and S. Christian, "Burden of care and quality of life amongst parents of children with seasonal allergies," *Journal of Allergy and Clinical Immunology*, vol. 103, pp. S72-S72, Jan 1999.

[4] C. Moller and S. Elsayed, "SEASONAL-VARIATION OF THE CONJUNCTIVAL PROVOCATION TEST, TOTAL AND SPECIFIC IGE IN CHILDREN WITH BIRCH POLLEN ALLERGY," *International Archives of Allergy and Applied Immunology*, vol. 92, pp. 306-308, Dec 1990.

[5] E. Kurt, S. Metintas, I. Basyigit, I. Bulut, E. Coskun, S. Dabak, F. Deveci, F. Fidan, H. Kaynar, E. K. Uzaslan, K. Onbasi, S. Ozkurt, G. Pasaoglu, S. Sahan, U. Sahin, K. Oguzulgen, F. Yildiz, D. Mungan, A. Yorgancioglu, and B. Gemicioglu, "Prevalence and risk factors of allergies in Turkey: Results of a multicentric cross-sectional study in children," *Pediatric Allergy & Immunology*, vol. 18, pp. 566-574, 2007.

[6] C. Penard-Morand, D. Charpin, C. Raherison, C. Kopferschmitt, D. Caillaud, F. Lavaud, and I. Annesi-Maesano, "Long-term exposure to background air pollution related to respiratory and allergic health in schoolchildren," *Clinical and Experimental Allergy*, vol. 35, pp. 1279-1287, 2005.

[7] Y. L. Lee, C. K. Shaw, H. J. Su, J. S. Lai, Y. C. Ko, S. L. Huang, F. C. Sung, and Y. L. Guo, "Climate, traffic-related air pollutants and allergic rhinitis prevalence in middle-school children in Taiwan," *European Respiratory Journal*, vol. 21, pp. 964-970, Jun 2003.

[8] J. Bartkova-Scevkova, "The influence of temperature, relative humidity and rainfall on the occurrence of pollen allergens (Betula, Poaceae, Ambrosia artemisiifolia) in the atmosphere of Bratislava (Slovakia)," *Int J Biometeorol*, vol. 48, pp. 1-5, 2003.

[9] R. S. Knorr, S. K. Condon, F. M. Dwyer, and D. F. Hoffman, "Tracking Pediatric Asthma: The Massachusetts Experience Using School Health Records," *Environmental Health Perspectives*, vol. 112, pp. 1424-1427, 2004.

[10] H. O. P. H. S. C. w. S. Mandate, "Highmark Outreach Program Helps Schools Comply with State Mandate," *Early Intervention.*, vol. 36, 2008.

[11] W. A. Yasnoff, P. W. O'Carroll, D. Koo, R. W. Linkins, and E. M. Kilbourne, "Public health informatics: improving and transforming public health in the information age," *Journal of public health management and practice : JPHMP*, vol. 6, pp. 67-75, 2000-Nov 2000.

[12] C. B. Gable, "A COMPENDIUM OF PUBLIC-HEALTH DATA SOURCES," *American Journal of Epidemiology*, vol. 131, pp. 381-394, Mar 1990.

[13] A. Friede and P. W. O'Carroll, "CDC and ATSDR electronic information resources for health officers," *Journal of public health management and practice : JPHMP*, vol. 2, pp. 10-24, 1996 1996.

[14] C. f. t. S. o. t. F. o. P. H. Institute of Medicine, *The Future of Public Health*. Washington DC: National Academy Press, 1988.

[15] V. Schoenbach and W. Rosamond, Eds., *Understanding the Fundamentals of Epidemiology: an evolving text.* Chapel Hill: University of North Carolina, 2000, p.^pp. Pages.

[16] U. W. I. Systems. (2011, June 23). *Upper Air Data*. Available: http://weather.unisys.com/upper_air/details.php

[17] G. Fernandez, *Data Mining Using SAS Applications*. Boca Raton, FL: Chapman & Hall/ CRC, 2003.

[18] D. Boos, "On Generalized Score Tests," *The American Statistician*, vol. 46, pp. 327-333, 1992.

[19] A. Rotnizky and N. Jewell, "Hypothesis Testing of Regression Parameters in Semiparametric Generalized Linear Models for Cluster Correlated Data " *Biometrika*, vol. 77, pp. 485-497, 1990.

[20] N. C. f. E. Statistics. (2012, February 22, 2012). *School District Demographic System - Map Viewer*. Available: http://nces.ed.gov/surveys/sdds/ed/index.asp?st=PA

[21] F. Highmark, "eTools Schools: Demographic Characteristics (Dataset)," I. InnerLink, Ed., ed. Lancaster, Pennsylvania: eTools, 2011.

[22] J. N. Logue, M. V. White, and D. J. Marchetto, "Pennsylvania's Asthma School Project and Descriptive Pilot Investigation: A focus on Environmental Health Tracking," *Journal of Environmental Health*, vol. 70, pp. 21-27, 2007.

Optimization and Evaluation Criteria in GP Regression

Rikard König[1], Ulf Johansson[1], Lars Niklasson[2]
[1]University of Borås, [2]University of Skövde, Sweden
rikard.konig@hb.se, ulf,johansson@hb.se, lars.niklasson@his.se

Abstract— **Although the use of genetic programming (GP) for predictive modeling in data mining increases, a large majority of these projects still focus on classification. We argue, however, that the abilities inherent in genetic programming, most notably the possibility to choose both representation language and optimization criterion based on the specific problem, must be even more important for predictive regression. Most importantly, while classification models are normally evaluated using accuracy, there is a large number of error metrics used for evaluating regression models. In this paper, we evaluate the use of five existing error metrics as fitness functions in GP regression. The overall purpose is to investigate how the optimization of the different error metrics affects the produced model. Specifically, in the experimentation, all models, regardless of the fitness function used, are evaluated using all error metrics. The main result, obtained on 16 UCI data sets, is that GP models evolved using a specific error metric, when evaluated using the same metric, generally obtained lower errors than both other GP models and models produced by two specialized regression tree techniques. This result shows that it is possible for a data miner to choose which aspect of a predictive regression model that should be prioritized, just by using GP and a suitable fitness function. Finally, when considering all error metrics, a fitness function minimizing the mean absolute error was shown to produce the most robust models. .**

I. INTRODUCTION

The purpose of predictive modeling is to utilize stored data in order to predict an unknown (often future) value of a specific variable; the target variable. When using machine learning techniques, the resulting model represents patterns in historical data found by the specific algorithm. More technically, the algorithm uses a set of training instances, each consisting of an input vector $\mathbf{x_i}$ and a corresponding target value y_i to learn the function $y=f(\mathbf{x};\Theta)$. During training, the parameter values Θ are optimized, based on a score function. When sufficiently trained, the predictive model is able to predict a value $\hat{y}$, when presented with a novel (test) instance $\mathbf{x}_j$. If target values are restricted to a predefined number of discrete (class) labels, the data mining task is called *classification*. If the target variables instead are real numbers, the task is named *regression*.

When performing predictive classification, the primary goal is to obtain high accuracy; i.e., few misclassifications when the model is applied to novel data. For regression problems, however, it is not obvious how predictive performance should be evaluated. Specifically, unlike classification, there exists a number of reasonable error metrics for regression. As a matter of fact, Armstrong [1] even argues that since these different metrics evaluate different aspects of the model's predictive performance, a model should preferably be evaluated using several measures. This is of course a very sensible advice in theory, but in real-world problems, the demands on the predictive model, if carefully analyzed, often indicate that one error metric is most suitable for the particular problem. Hence, which error metric to use for the model evaluation appears to be a key decision for regression problems.

In addition, the business domain of the regression problem often imposes even further demands on the evaluation criterion. As an example, in the retail domain it is most often considered a more serious error to underestimate a demand than to overestimate it. This is actually a very serious problem during campaigns or promotions, the effects of which are notoriously hard to predict, where poor forecasts may lead to very dissatisfied customers when the targeted merchandise runs out of stock.

In practice, the root mean square error (RMSE) or the mean square error (MSE) is often favored due to their mathematical aspects, i.e. MSE is additive for independent sources of distortions, symmetric, convex, differentiable and distance preserving under orthogonal and unitary transformations [2]. Another important property often mentioned as the main reason for using these measure is that they penalize larger errors more severely. RMSE or MSE are, however, far from obvious choices. Armstrong even argues that they are some of the worst metrics since they are poorly protected against outliers and scale dependent, i.e. errors from two datasets cannot be compared if the dependent variables of the datasets do not have the same scale.

Most importantly, it must be noted that although a predictive model can be evaluated against any number of error metrics, it is normally optimized against one specific metric during the construction. Specifically, the optimization criterion (the score function) is actually inherent in most modeling techniques. For example, *Multiple linear regression*, *Auto Regressive Moving Average* and *Exponential smoothening* all minimize the mean square error, i.e., these techniques will all suffer from the problems associated with metrics like MSE and RMSE.

Regression trees are also often optimized based on similar measures; Quinlan's M5 [3] does for example use the standard deviation as optimization criteria and the RMSE during pruning.

Naturally, optimizing a specific metric will produce a model focused on that criterion. With this in mind, it becomes important to understand how different metrics correlate; specifically how using one metric for the optimization will affect the quality of the model, possibly also evaluated using another metric. One obvious question is, of course, if there is a specific metric that when used as the optimization criterion will produce models that rank high on all evaluation criteria. If this is not the case, data miners are forced to choose very early in the modeling process exactly what the final model should prioritize, making the choice of technique and error metric even more important.

A related and important point, stressed by Armstrong, is that it very questionable if the data miners should be responsible for determining how the errors should be interpreted and evaluated. Even if a larger (squared) penalty for a larger error makes sense in the forecasting setting, it is not sure that the domain expert is of the same opinion.

From this reasoning, it is very interesting, but also slightly discouraging, that the choice of optimization criterion is often neglected in descriptions of data mining processes. Even CRISP-DM [4], a cross industry standard for the data mining process, fails to address this important issue. CRISP-DM only states that any assumptions that are implicit in the modeling technique should be defined explicitly, but does not even mention that the optimization criterion would have an effect on the resulting model. CRISP-DM does, however, stress that most real-world data mining projects eventually will be evaluated using monetary measures, thus acknowledging the effect the business context has on the choice of optimization criterion. Or, put in another way, which optimization criterion to use during the model building must ultimately be analyzed for each and every project, in order to make sure that it really agrees with the overall purpose of the project.

From the description above, it is obvious that there is a need for regression techniques making it straightforward to pick or change the optimization criterion. Naturally, evolutionary algorithms have exactly this property since the role of the fitness function makes it easy to directly encode any optimization criterion. As a matter of fact, evolutionary optimization is of course not even limited to existing error metrics, but could use score functions tailor-made for the specific problem. As an example, it would be quite straightforward to penalize underestimated and overestimated demand differently, in order to handle the retail sales problem discussed above.

In addition, using Genetic Programming (GP), would give the data miner the opportunity to not only design the optimization criterion, but also the models' representation language, i.e., GP could, in theory, apply an arbitrary optimization criterion, to any kind of (standard or not) regression model. With this in mind, it is quite surprising that while GP classification, in a data mining framework, has received a lot of interest lately, there are very few GP studies explicitly targeting the potentially even more suitable regression task.

The overall purpose of this paper is to investigate the use of different optimization criterion as fitness functions in GP regression. Specifically, we analyze how the choice of score function affects the model quality, evaluated using not only the metric encoded in the score function, but also several others. In this study, we evaluate only well-established error metrics, and we restrict the models to *regression trees*. In the experimentation, the GP produced trees are compared not only to each other, but also to two existing standard techniques, *REPtrees* and *M5P*.

II. BACKGROUND

Keijzer [5], gives an excellent example of the difficulties that can occur when using a straightforward GP approach to a symbolic regression problem. Keijzer evaluates the GP success rate for two simple target functions $t=x^2$ and $t=100+x^2$. The squared error was used as fitness function and 50 GP runs over 20 generations were performed for each function. The first function was often found already in the first generation and had a success rate of 98%; the second function however was seldom found and had a success rate of only 16%. Keijzer argues that the low success rate is due to that the selection pressure enforces the GP process to spend most of its effort on getting the range of the constant right. Once found the diversity is so low that the square function is never found. Hence, most of the runs for the second function converge on the average of the training data and only eight runs managed to find something better.

Keijzer then showed how this problem, called the *range problem*, can be eliminated for polynomial expressions by scaling the output of each program with a simple linear regression. Given that p_i is the prediction of the program for the instance i the prediction can be scaled using equation (1), where the slope k and the intercept m for the linear regression are calculated beforehand, using the least square method.

$$p_{i_{scaled}} = k * p_i + m \quad (1)$$

For regression trees the same problem of finding the correct range would appear in each leaf node. It would, however, not help to scale the final output of the model, since the range of the intercept would need to be in a scale appropriate for the associated variable.

In the light of the range problem, as demonstrated by Keijzer, the most straightforward approach to evolving model trees, even with simple regressions based on a single variable, will experience severe problems. Nevertheless this approach has of course been applied by numerous researchers with varying amount of success.

Another approach to handling the range problem was taken in [6], where regression trees were evolved using a fitness

function based on the mean absolute error (MAE). The initial population was created using an algorithm similar to CART [7]. To achieve diversity, each individual was created using randomly selected subsets of the training data. Since the ephemeral constants were created using the decision tree algorithm, they would always be in an appropriate range. In that study, mutation was also performed by calculating a new split with the same method. The proposed technique was evaluated against WEKA's [8] REPTree on 14 UCI datasets. Surprisingly, the results were rather modest with seven wins and seven losses when evaluated using RMSE. The evolved trees were however slightly smaller (i.e., easer to interpret) than the trees induced by REPTree.

Finally it should be noted that this and similar methods automatically makes the constants dependent on the criteria optimized by the decision tree algorithm, e.g., least square or least deviation. So, the constants were initialized based on RMSE, the tree was optimized on MAE, and the final model was evaluated on RMSE. Clearly, the implicit choice to mix these error metrics, especially evaluating on a metric not targeted during the model construction, may very well be the reason for the somewhat discouraging results.

A. Error Metrics

The following section presents five of the most common error metrics for regression tasks. They all have their own advantages and disadvantages and are appropriate for different task.

1) Mean Absolute Error

The most straightforward estimate of the error of a regression model is the *mean absolute error* (MAE). It is easy to calculate and easily understood by managers and decision makers. It is however scale dependent which, for instance, make comparison over several datasets cumbersome. The MAE for a model m can be calculate using equation 2 below, where m_i is the prediction made by the model for the instance i and a_i is the actual value for the same instance.

$$MAE_m = \frac{\sum_{i=1}^{n} |m_i - a_i|}{n} \quad (2)$$

2) Root Mean Square Error

The *mean square error* (MSE) and *root mean square error* (RMSE) two of the most frequently used metrics today, especially by statisticians. One motivation for the use of RMSE is, as mentioned in the introduction, that it has several desirable mathematical properties but also that it emphasizes eliminating large errors. Armstrong does, however, point out several deficiencies of RMSE; it is, as mentioned in the introduction, scale dependent just like MAE, and very sensitive to outliers.

$$RMSE_m = \sqrt{\frac{\sum_{i=1}^{n} (m_i - a_i)^2}{n}} \quad (3)$$

In some cases it could of course be appropriate to use RMSE, but this decision must be taken after discussing the specific problem with domain experts, i.e., RMSE should not be used routinely just because the data miner selected a modeling technique implicitly optimizing RMSE.

3) Mean Absolute Percentage Error

Another popular metric is the *mean absolute percentage error* (MAPE), which strongest advantages are that it is easy to interpret and that it is scale free. A problem with this measure is, however, that it is biased towards low predictions for problems which do not include negative target values. Problems with only positive target values are of course very common, e.g., when predicting demand or actual sales. In these cases, even a prediction of 0 would result in an error of no more than 100%, while a too high prediction could potentially result in an enormous error. Another problem is that this measure is undefined when the target value is zero. To handle these drawbacks a cap for the maximum error (Max_e) of a single error is often used, see equation (3). Nevertheless, for these reasons MAPE is not a good metric for problems where the target values may be close to zero.

$$MAPE_m = \frac{\sum_{i=1}^{n} min\left(\left|\frac{m_i - a_i}{a_i}\right|, Max_e\right)}{n} \quad (4)$$

4) Mean Unbiased Absolute Percentage Error

Due to the deficiencies identified for MAPE, as described above, Makridakis [9] instead argues for the use of the *Mean Unbiased Absolute Percentage Error* (MUAPE), since it is unbiased and is less likely to have troubles when the targets values are close to zero. According to Makridakis, MUAPE is still comprehensible for decision makers, but this has been heavily disputed.

Even if MUAPE is more robust for predictions around zero it still needs to be capped, similar to MAPE.

$$MUAPE_m = \frac{\sum_{i=1}^{n} min\left(\left|\frac{m_i - a_i}{(m_i + a_i)/2}\right|, Max_e\right)}{n} \quad (5)$$

5) The Pearson Correlation

The Pearson correlation coefficient, often denoted as r, is yet another frequently used measure. The Pearson correlation, of course, measures the strength of the linear dependencies between the predicted and target value. Often the square of r is used instead, thus illustrating how much of the variability of the target variable that is explained by the model. Although both uses of the Pearson correlation coefficient convey important information of the model's predictive performance, it must be noted that they ignore the sizes of all errors. The Pearson correlation should therefore rarely be used as the sole criterion.

$$r_m = \frac{\sum_{i=1}^{n} \left((m_i - \overline{m_i})(a_i - \overline{a_i})\right)}{\sqrt{\sum_{i=1}^{n} (m_i - \overline{m_i})^2 * \sum_{i=1}^{n} (a_i - \overline{a_i})^2}} \quad (6)$$

III. METHOD

A. Proposed GP implementation

As described in the background, using scaled ephemeral constants greatly increases the chances of finding good polynomial expressions or regression trees. For the regression tree representation, which is the focus of this study, one straightforward solution would of course be to use the average value of all training instances reaching a specific leaf as the leaf constant. The average value would, however, only be optimal if RMSE was selected as the criterion to minimize. For other metrics such as MAE, MAPE or MUAPE, a different leaf value could very well be optimal. Consider for example a leaf node which observes three training instance with the target values 1, 2 and 4; the average value 2.33 would result in the minimum error according to RMSE. However, as can be seen in Table 1, it would, according to MAE and MUAPE, be better to predict the value 2 while a prediction of 1 would be optimal for MAPE.

Predicted Value	MAE	RMSE	MAPE	MUAPE
2.33	1.11	**1.25**	82.1%	49.3%
2	**1.00**	1.29	64.3%	**44.4%**
1	1.33	1.83	**53.6%**	62.2%

Table 1 - Prediction errors for values (1,2,4)

As demonstrated by this simple example, using mean values as leaf constants is only optimal if using RMSE. Or, put the other way around, the method to find good ephemeral constants is dependent on the optimization criterion, and, specifically, using mean values is actually suboptimal if any other metric than RMSE is used.

It may be noted that it would of course be possible to calculate the optimal value for each measure, but this would restrict the generality of the GP process. If any changes were incorporated in the fitness function, like adding weights or introducing a new measure, the leaf class would have to be updated to produce the optimal value for the new fitness function.

In this study, a more flexible approach is instead taken to let the evolution find the right constants, but then scaling the ephemeral constants to the range of the training instances reaching each node. Internally all leaf constants C_i are initialized to a random value between zero to one. These internal constants are however scaled by the maximum and minimum value of the dependent variable *y* for the training instance reaching the leaf *l* according to equation 7.

$$C_{rnd} = \min(y_l) * C_i * +(1 - C_i) * \max(y_l) \quad (7)$$

This approach drastically reduces the range problem while, at the same time allowing optimization of an arbitrary fitness function. Another advantage is that the general meaning of the internal constants are conserved through crossover, i.e. a high value will result in a prediction near $\max(y_l)$ in all leaves, even if $\max(y_l)$ will be different in different leaves.

B. Techniques

To enable comparison with the evolved regression trees, two standard techniques also producing regression trees were included in the experimentation. For this benchmarking, *REPTree* and *M5P*, as implemented in the WEKA workbench [8], were chosen. These techniques, and the GP framework used, are presented in more detail in the following sub-sections.

1) G-REX

G-REX is our open source GP framework for data mining [10,11]. Two important features of G-REX are the ability to optimize arbitrary error measures in the fitness functions, and the fact that the representation language, including the syntax, can be defined externally.

The error measures targeted in this study have been implemented into five different fitness functions, which general form can be described by equation (8). *M* is the optimized measure, *p* is the program under evaluation, O_p is the complexity of *p* (nodes + leaves) and P_1 is the coefficient for the parsimony pressure.

P_2 is the coefficient for a second parsimony pressure aimed to facilitate evolution of programs of a certain complexity. Since the complexity of a tree inherently decides how many different values it is able to predict, a higher complexity may lead to an unfair advantage. Hence, it is in this setting desirable if all trees had roughly the same complexity, since then it is actually the choice of ephemeral constants that affects have the greatest effect on the performance. To enforce this, the length penalty P_1 is typically set to a small value and P_2 to a much larger value. Hence, a large punishment will be added for each extra element above a maximum complexity threshold O_{max}.

MAPE, MUAPE and *r* are scale independent, but MAE and RMSE must be made scale independent by dividing with the mean of the actual value according to equation (9). Scale independence of course ensures that the same level of pressure will have similar effect on the evolution, thus minimizing the amount of parameter tuning. Still, the effect may vary slightly since each metric represents a different error function, i.e., the change in deviation does seldom affect the error the same way for all measures. Finally *r* values were replaced with *1-r* to facilitate minimization of all fitness functions

$$f_{M_p} = M_p + P_1 * O_p + P_2 * \max(O_p - O_{max}, 0) \quad (8)$$

$$f_{M_p} = \frac{M_p}{\bar{a}} + P_1 * O_p + P_2 * \max(O_p - O_{max}, 0) \quad (9)$$

Since the optimization criteria encoded in the fitness functions, were made scale independent and equal in range, the same set of parameters, found by initial experimentation, could be used for all fitness functions.

For this study, G-REX used the same representation language as REPTree and *M5P*. Table 2 below shows the BNF used by G-REX and figure 1 shows a sample tree created for the *Machine_CPU*, data set.

Functions	{ *if*, <=, >, =}
Terminals	{ *catV*, *conV*, C_{rnd}, C_{conV} , C_{catV} }
program	::= *expression*
expression	::= *if-statement* | C_{rnd}
if-statement	::= if *condition* then *expression* else *expression*
condition	::= $conV$ <= C_{conV} | $conV$ > C_{conV} | $catV$ = C_{catV}
C_{rnd}	::= Constant initialized according to equation 7.
C_{conV}	::= Constant initialized to a random value of *conV*
C_{catV}	::= Constant initialized to a random value of *catV*
conV	::= Independent continuous variable
catV	::= Independent categorical variable

Table 2 – BNF for the regression trees optimized using GP

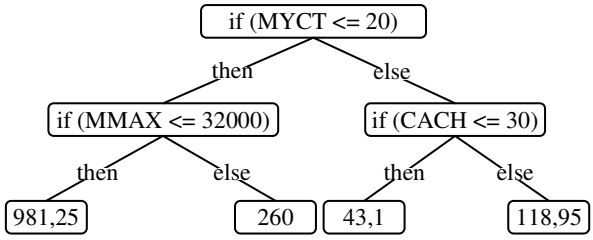


Figure 1 – Sample regression tree for Machine_CPU

2) REPTree

REPTree is a decision tree technique implemented in the WEKA workbench that is optimized for speed. It builds trees by reducing the variance of the resulting subsets T_1 , T_2 of each split (VarR) according to equation 8.

$$VarR = Var(T) - \sum_i \frac{T_i}{T} * Var(T_i) \qquad (10)$$

By default, 2/3 of the training data are used to build the tree and 1/3 for *reduced error pruning*, a post-processing technique which prunes a tree to the subtree which have the lowest squared error (SE) on the pruning data. Leaf constants are calculated as the mean value.

3) M5P

M5P is another model tree technique (based on Quinlan's M5 [3]) implemented in the WEKA framework. M5P first grows an optimal decision tree using all training data by minimizing the standard deviation and then prunes it to minimize the expected RMSE of the tree. Since the tree is optimized on the training data it will underestimate the true error of the tree. To compensate for this, the expected error is calculated by multiplying the RMSE in each leaf with a factor (n+v)/(n-v), where *n* is the number of instances reaching that leaf and *v* is the number of parameters of the model. Hence, the expected error will be calculated with larger pruning factors for large tree since they naturally will contain leaves with fewer instances. The final pruned tree is the subtree with the lowest expected error. When used to created regression trees, the leaves predicts the mean value of the training instance reaching the leaf.

M5P also has the option to use smoothing, i.e., using the average of a linear model created in each node leading to a certain leaf, but this is not used here.

C. Experiments

The overall aim of this study is to investigate how the optimization of different error measures affects the produced model. To do this, all models, regardless of which technique that was used to create it, are evaluated against all error metrics; i.e., MAE, RMSE, MAPE, MUAPE and *r*. Since both MAPE and MUAPE have problems when the target values are close to zero, they are trimmed by setting the Max_{error} in equation 4 and 5 to 10. Since neither MAE nor RMSE is scale free, they are normalized according to equation 7.

For the actual evaluation, standard stratified 10-fold cross-validation was used; i.e., each technique was used to create one regression tree per fold. The result for a specific setup and data set is of course averaged over all folds.

In this study, G-REX creates regression trees using the BNF presented in Table 2 for each of the five fitness functions, as described in section 3. To achieve a more robust performance of the GP optimization, a batch of three separate runs were performed for every fold in each dataset. The tree with the best fitness of the three batches was selected as the final tree, and used for evaluation. Identical GP parameters, as presented in Table 3 below were used for all fitness functions and datasets.

GP Parameter	**Value**
Population Size	*1000*
Generations	*100*
Creation Type	*Ramped Half and Half*
Crossover Probability	*0.8*
Mutation Probability	*0.01*
Complexity Threshold (O_{max})	*15*
Parsimony pressure (P_1)	*0.01*
Parsimony pressure (P_2)	*0.5*
Batches	*3*

Table 3 – GP Settings

Dataset	Size	Num	Nom	Mean	Std
auto_price	159	14	1	11446	5878
baskball	96	5	0	0.42	0.11
bolts	40	8	0	20.5	11.69
cholesterol	303	7	7	246.69	51.78
cloud	108	5	2	1.23	1.08
gascons	27	5	0	207.03	43.79
housing	506	13	1	22.5	9.2
machine_cpu	209	6	0	105.6	161
pharynx	195	2	10	555.5	422
pollution	60	16	0	940.36	62.21
quake	2178	4	0	5.98	0.19
sensory	576	1	11	15.08	0.82
servo	167	1	4	1.39	1.56
sleep	62	7	0	8.5	5.8
veteran	137	3	4	121.6	158
wisconsin	194	32	0	46.9	34.5

Table 4 –Properties of datasets

Sixteen regression datasets from the UCI-Repository [12] was used in the experiments. Table 4 above presents the number of instances *size*, the number of numeric and nominal variables *Num* and *Nom,* and the *mean* and the standard deviation *Std* of the dependent variable.

IV. RESULTS

Table 5 shows the average ranks over all data sets for each technique and each metric. Results in bold signify the lowest rank obtained among all techniques, while the best result among the different GP fitness functions are underlined.

One first observation is that the fitness functions work as intended. On the training set, the GP optimizing that specific criterion is always the best setup overall. Another interesting pattern is that on the training data, GP using MAE_Fitness actually outranks both regression tree techniques on all performance metrics. In addition, that setup is also almost always better than all other GP setups not specifically targeting the criterion used for the evaluation. This of course indicates that using MAE in the fitness function will produce models that also have quite good results on the training data according to the other metrics.

Looking at the test results, the general picture is that optimizing a specific criterion on the training data will produce models that rank well when that same criterion is used for the actual evaluation. GP using MAE_Fitness, GP using MAPE_Fitness and GP using CORR_Fitness all obtained the best rank overall on the test data when the respective criterion was used for the evaluation. On RMSE, GP using RMSE_Fitness was clearly the best among the GP setups, but REPtree was even slightly better.

On MUAPE, however, GP using MUAPE_Fitness was outperformed by both regression tree techniques and by GP using MAE_Fitness. A possible explanation to this phenomenon is that MUAPE is measured in percent, which makes the measure less precise, especially since the percentage is affected by the levels of both the predicted and the actual value. Based on this, a model optimized on this criterion, may appear to be overly specialized; i.e., will generalize poorly. This is also apparent in the results, where GP using MUAPE_Fitness has the best MUAPE result on the training data, but among the worst on the test data.

MAPE, which also define the error in percent, does however not suffer from this problem, which can be explained by the fact that MAPE is only metric biased towards low estimations.

Comparing GP to the specialized techniques, GP appears to be the best option overall, but REPTree, as mentioned above obtained slightly better RMSE. This is despite the fact that GP using RMSE_fitness had much lower training rank. It must, however, also be noted that the tree produced by the REPTree technique are almost twice as large.

Given the fact that the size of the GP trees was restricted to facilitate a fair comparison between the different GP setups, the comparison against M5P is arguably more interesting since M5P trees are of more similar size. M5P does however perform quite similar to REPTree when compared to the GP setups. The only difference that shows up in the ranks is that RMSE_Fitness beats M5P but not REPTree. This is probably a result explained by that M5P trees are less complex which translates to fewer separate prediction values. Hence, M5P models and the GP models (which have similar sizes) cannot be fitted to as many extreme values as the REPTrees. Since RMSE punish larger errors harder this restriction makes the less complex models inferior. All other metrics gives an error that is proportional to size of the deviation and are thus less affected by the size of the models. This can also be seen as REPTree and M5P perform very similarly on all other measures.

Finally, it is very interesting to compare all models to see which approach that gave the most robust model, i.e., had the best result considering all of the evaluated metrics. To make this comparison as fair as possible, the mean ranks presented in Table 5 below were averaged over all evaluation metrics to produce an overall ranking for each technique on training and test, see Table 6.

Technique	MAE		RMSE		MAPE		MUAPE		R		Size	
	TRN	TEST	TRN	TEST	TRN	TEST	TRN	TEST	TRN	TEST	Rank	Size
M5P-Reg-US	4.44	3.44	4.09	3.44	4.63	4.03	4.13	3.28	4.94	4.44	**2.81**	**11.50**
REPTree	3.84	3.63	3.75	**3.31**	4.03	4.03	3.41	**3.25**	4.81	4.19	4.56	21.40
MAE_Fitness	**2.13**	**3.16**	3.09	3.78	3.25	3.63	2.84	3.34	3.50	4.06	3.97	13.55
RMSE_Fitness	3.34	3.91	**1.59**	3.34	5.09	4.47	4.66	3.78	2.19	3.75	4.56	13.76
MAPE_Fitness	4.59	4.25	5.72	4.69	**1.69**	**2.44**	4.06	4.47	5.53	4.50	3.69	12.88
MUAPE_Fitness	3.66	3.91	4.81	4.31	3.63	4.03	**2.53**	3.69	5.22	4.19	3.72	13.40
CORR_Fitness	6.00	5.72	4.94	5.13	5.69	5.38	6.38	6.19	**1.81**	**2.88**	4.69	14.53

Table 5 – Mean ranks

From the results presented in Table 6 it is obvious that GP using MAE_Fitness was clearly the most robust technique since it obtained the lowest overall rank on both training and test data. REPTree was the second best technique overall, but again it may be noted that the REPTrees where almost twice as large as the trees produce by all other techniques.

REG-Trees	TRAIN	TEST
M5P-Reg-US	4.80	3.40
REPTree	3.60	2.70
MAE_Fitness	**2.00**	**2.60**
RMSE_Fitness	3.40	3.90
MAPE_Fitness	5.00	5.20
MUAPE_Fitness	3.60	4.40
CORR_Fitness	5.60	5.80

Table 6 – Mean overall ranks over all measures

V. CONCLUSIONS

We have in this study evaluated the use of five well-known error metrics as fitness functions in GP regression. In addition, we have evaluated all produced GP models, using the same five error metrics, on the test data. Finally, the predictive performance of the GP regression trees have also been compared to two standard specialized regression tree techniques.

The main result is that GP models evolved using a specific error metric, when evaluated using the same metric, always obtained the best result on the training data, and most often also on the test data. This is of course a reassuring result, making it possible for a data miner to choose a fitness function to ensure that the produced model prioritizes the intended property.

Comparing the evolved models to the models produced by the specialized regression tree techniques, the results show that when using the corresponding fitness function, the GP outperformed both regression tree techniques on MAE, MAPE and *r*. On RMSE, REPTree obtained the best mean rank, but the GP trees evolved based on the RMSE fitness had very similar results and were almost half as complex. Only when using MUAPE for the evaluation was the GP approach clearly outperformed by the specialized techniques.

Finally, when investigating how models evolved using one specific fitness fared when evaluated using all error metrics, the MAE fitness was shown to produce the most robust models.

VI. DISCUSSION AND FUTURE WORK

In this paper, only existing evaluation metrics were targeted. Still, it should be noted that the evolutionary framework allows tailor-made optimization and evaluation criteria. In addition, multi-objective optimization could be implemented rather easily. As an example, it would be interesting to combine one measure like MAE or RMSE with a correlation based measure. With this in mind, we intend to, in future studies, suggest and evaluate error metrics specific for different data mining situations.

REFERENCES

[1] Armstrong, J.S. 2001. *Principles of forecasting: a handbook for researchers and practitioners*. Kluwer Academic Publishers.

[2] Wang, Z. and Bovik, A.C. 2009. Mean squared error: Love it or leave it? A new look at signal fidelity measures. *Signal Processing Magazine, IEEE*. 26, 1 (2009), 98–117.

[3] Quinlan, J.R. 1992. Learning with continuous classes. *5th Australian joint conference on artificial intelligence* (1992), 343–348.

[4] Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C. and Wirth, R. 1999. *CRISP_DM 1.0 Step-by-step data mining guide*. CRISP DM Consortium.

[5] Keijzer, M. 2003. Improving symbolic regression with interval arithmetic and linear scaling. *Genetic Programming*. (2003), 275–299.

[6] Kretowski, M. and Czajkowski, M. 2010. An evolutionary algorithm for global induction of regression trees. *Artifical Intelligence and Soft Computing: Part II* (2010), 157–164.

[7] Breiman, L. 1984. *Classification and regression trees*. Chapman & Hall/CRC.

[8] Witten, I.H. and Frank, E. 2005. *Data mining: practical machine learning tools and techniques*. Morgan Kaufmann.

[9] Makridakis, S. 1993. Accuracy measures: theoretical and practical concerns. *International Journal of Forecasting*. 9, 4 (1993), 527–529.

[10] Johansson, U., König, R. and Niklasson, L. 2003. Rule extraction from trained neural networks using genetic programming. *International Conference on Artificial Neural Networks and International Conference on Neural Information Processing* (Istanbul Turkey, 2003), 13-16.

[11] König, R., Johansson, U. and Niklasson, L. 2008. G-REX: A Versatile Framework for Evolutionary Data Mining. *IEEE International Conference on Data Mining Workshops, 2008. ICDMW'08* (2008), 971–974.

[12] Blake, C. and Merz, C. 1998. UCI repository of machine learning databases. (1998).

AUTOMATED PROVISIONING OF CAMPAIGNS USING DATA MINING TECHNIQUES

Saravanan M
Ericsson R&D
Ericsson India Global Services Pvt. Ltd
Chennai, Tamil Nadu, India
m.saravanan@ericsson.com

Deepika S M
Department of Information Technology
Thiagarajar College of Engineering
Madurai, Tamil Nadu, India
deepikatce@gmail.com

Abstract – **Today's telecom industry needs to use customer information for carrying out analysis, design and execution of new campaigns. The present and the traditional campaign generation process has identified few broad segments of customers based on some generalized information and their interaction with such segments. Automation in the process make it easy for the marketers to monetize data, increase customer life time value and to fill up gaps in existing systems. Using data mining techniques in telecom industry helps operators to tune interactive relationships to each of the segment needs. In this paper, we analyze the customer CDRs (Call Detail Records) details to expedite feature extraction on their behavioral aspects. Suitable data mining algorithms have been employed on the extracted features to understand the customer behavior and the likelihood that a user will accept the campaign. Usually the number of campaign takers are very less when compared to other large collection. This leads to the problem of class imbalance. Different combinations of techniques have been used to overcome it. Thus we find out the importance of the customer features for the different types of campaigns and the contributing features for each type of campaign. The customers who exhibit certain features have been selected for specific type of campaign. We have evaluated the use of different data mining classification techniques and the results obtained demonstrate that the Cost Sensitive Stacking method performs comparatively better than other individual classification methods related to customer selection for specific campaigns.**

I. INTRODUCTION

Telecom Industry provides the preliminary means of communication and a number of related services to the customers. One of the important procedures is the way they communicate the services by generating campaigns to customers. Campaign Management process can be described as a business driver which uses marketing tactics to achieve a particular business goal. The business goal is to obtain a higher response rate from the customers for the campaigns. The process of marketing campaign optimization takes a set of offers to customers and determines the offers that should be provided to a particular set of customers at a specific time.

Automation in the campaign management process allows marketers to create, test, execute, and report on multiple campaigns direct from the marketing desktop, without any need for complex scripting and with minimal human involvement [1]. It also helps to overcome the existing gaps such as paying less attention to existing customers, absence of a method to select the appropriate group of customers for launching campaigns and measurement of the response of the customers for previous campaigns. Automated process facilitates the building of new campaigns depending upon the features, which reduces the time to a large extent. Since automated campaigns selects a group of customers depending upon their previous behaviors for launching, the problems of losing focus on existing customers and proper measurement of response are avoided. Automation process uses data mining techniques to find the specific group of customers to launch campaigns [2].

Data mining in telecom domain is generally used to analyze the large databases and provide meaningful results. It plays a significant role in recognizing and tracking patterns within the data. The advantage of using data mining techniques is that it even extracts information from the databases which may not be known as existed [3]. From our initial experimental studies, we found that some of the relevant data mining classification algorithms such as Cost Sensitive classification [4], OneR rule [5], Bayes Net [6], J48 Decision Tree, Logistic regression [7] and Combined Regression and Ranking method (CRR) [8] performs well compared to the other classification algorithms like SVM, gradient boosting, etc. These algorithms were employed in this study. Classification algorithms generally faces class imbalance problem when worked with skewed data samples. Class imbalance occurs when the number of instances representing a particular class is very less when compared to the remaining instances (other class). The classification algorithms, when trained with class imbalance data, generate poor results for the corresponding test set.

The proposed method analyses the customer behavior from the CDRs and extracts relevant features for different types of campaigns and thus selects the set of customers to launch a new campaign. It analyses the behavior from their previous responses to campaigns, importance to the existing customers is assured and their responses are measured.

In addition to the simple classification algorithms, voting and cost sensitive stacking methods using different combinations of those algorithms with different values for the cost matrix were trained on the training set and evaluated on the test set. Precision, Recall, F-Measure, Kappa statistic and Accuracy [9] are the different measures considered to evaluate the applied data mining algorithms. Also, we have applied

CRR to generate a better recall value.

The remainder of this paper is organized as follows. In Section 2, we give an overview of the areas related to this study. Section 3 explains the experimentation of various data mining algorithms in our approach. We provide an outline of our proposed system in Section 4. Section 5 deals with the various Evaluation measures. We discuss the results in Section 6. Section 7 discusses the importance of this application and the concluding remarks are given in Section 8.

II. RELATED WORK

A very few of the existing techniques uses the method of customer segmentation for campaign management. Intelligence value based customer segmentation method [10] investigates the customer behavior using a Recency, Frequency and Monetary (RFM) model and then uses a customer Life Time Value (LTV) model to evaluate the proposed segmented customers. This method also proposes to use Genetic Algorithms (GA) to select appropriate customers for each individual campaign.

Predicting mobile phone churners in the telecom industry uses a similar method of analyzing the features from the customer CDRs. To tackle the problem of churn prediction, the telecom operator needs to understand the behavior of customers, and classify the churn and non-churn customers, so that the necessary decisions will be taken before the probable churner switch to a competitor. One of the solutions adapted for this problem is building up an adaptive and dynamic data-mining model in order to efficiently understand the system behavior and allow time to make the right decisions [11]. Our problem is similar to that of churn prediction problem where there is a need to tackle the imbalance in model usage. The churn prediction problem attached to telecom domain is thoroughly studied with different machine learning techniques such as Bayesian networks, association rules, decision trees and neural networks [11].

Evaluation in the customer segmentation method for campaign management uses a customer Life Time Value (LTV). This method takes the customer acquisition, customer development and customer retention stages into account [10]. Hybrid classification methods have been used for evaluation in churn prediction for mobile telecom data [12]. Since the number of churners is very less when compared to the non-churners, the existing classification models suffers from the problem of imbalance. Similar problem is faced in our method as the number of campaign takers for each and every unique type of campaign is very less when compared to the other group of customers. To solve this problem, a hybrid framework which combines the results of two or more classifiers which boosts the performance of the models is used [12]. In this paper, we have used cost sensitive stacking method to combine the results of two basic classifier results. Cost sensitive stacking has been used for audio Tag annotation and Retrieval [13]. The co-occurrence of tags is considered to model the audio tagging problem as a multi-label classification problem. Cost sensitive multi-label classification is used to boost the performance of individual classifiers. Cost sensitive stacking combines the outputs of multiple independent classifiers for multi-label classification. [13]. CRR [8] uses stochastic gradient descent that makes the algorithm easy to implement, and efficient for use on large-scale data sets.

The number of customers to be handled is huge in telecom domain; manual work takes a long processing time for campaign generation. Also to help operators run effective marketing campaigns by leveraging subscriber information and external data to build target lists and then to execute them through multiple channels, automated provisioning of campaign process becomes essential. Automation also helps the service providers to respond quickly to changing customer behavioral trends for the immediate generation of advertisements [1].

III. USAGE OF DATA MINING TECHNIQUES

The proposed approach aims to identify the actual customers who would respond to the campaigns based on their previous behavior. The behavior of the customers can be obtained by analyzing the various attributes from the CDR files. The attributes chosen for analysis includes the preliminary and derived attributes, aggregated on a weekly basis. In the process of extraction of features and understanding the method of implementation, automation in campaign management is studied through various related techniques which are discussed here.

A. Feature Selection

Feature Selection is a technique which is used to reduce the number of features before applying a data mining algorithm. Irrelevant features may have negative effects on a prediction task. It is also used for enhancing generalization capability, speeding up learning process, and improving the model interpretability. Feature selection has been applied in fields such as multimedia database search, image classification and biometric recognition [14].

The usage of feature selection techniques can be generalized into three main categories [15]: embedded approaches where feature selection is part of the classification algorithm, (e.g. decision tree), filter approaches where features are selected before the classification algorithm is used and wrapper approaches where the classification algorithm is used as a black box to find the best subset of attributes. Filtering methods assumes that the feature selection processes are independent from the classification step. Wrappers usually provide better results, the price being higher computational complexity. Certain classification algorithms use embedded approach of feature selection by finding the information gain of the selected attributes [16]. In our method, embedded approach and Wrappers has been used as the feature selection techniques.

Depending upon the value of the features selected for every particular type of campaign, classification algorithms are applied to predict the response of the customers for the campaigns.

B. Classification Algorithms

Prediction of the response of the customers to campaigns requires employing a supervised learning paradigm. In a supervised learning approach, we have training data D containing N samples. Each training instance X is a vector of d attributes (i.e) $X= \{x_1,x_2,....x_d\}$ The attributes can be numeric or nominal in nature. Classification algorithms are used to build a model using the training set and apply the built model on the test data set to predict the class labels. Some of the classification algorithms which suited to our dataset are described in this section.

1) OneR Classification:

OneR [5], which is a "One Rule" classification, is a simple, yet accurate, classification algorithm that generates one rule for each predictor in the data, and then selects the rule with the smallest total error as its "one rule". To create a rule for a predictor, a frequency table is constructed for each predictor against the target. The total error calculated from the frequency tables is the measure of each predictor contribution. A low total error indicates a higher contribution to the predictability of the model.

To create a rule for an attribute, the most frequent class for each attribute value must be determined. The most frequent class is simply the class that appears most often for that attribute value. A rule is simply a set of attribute values bound to their majority class.

For certain type of campaigns, one particular attribute may act as an influencing attribute. For such models, OneR classifier performs well and provides good results. In our analysis, experiments are conducted for each and every attribute and comparison between them. Moreover the use of OneR method resulted in good evaluation measures for a particular type of attribute for a campaign.

2) Logistic Regression Method

Regression analysis is a technique to predict the value of a dependent variable by fitting a function with least error on the training data. Logistic Regression fits an S-shaped curve to the data [7]. This method estimates the average value of the target variable when the independent variables are held fixed. This model is a non-linear transformation of a linear regression model. It is also referred as Logit function. It makes use of one or more predictor variables that may be either numerical or categorical. A Logistic function always takes on the values between zero and one.

Binary logistic regression refers to the instance in which the criterion can take only two possible outcomes. Our approach aims to predict whether the customer would belong to the specific class in which he can take up the campaign or fall in to other class. Since it is a binary classification [11], binary logistic regression can be considered as an appropriate method for building the predictor model.

3) BayesNet Classification

A **Bayesian network** is a probabilistic graphical model (a type of statistical model) that represents a set of random variables and their conditional dependencies via a directed acyclic graph(DAG).

To use a Bayesian network as a classifier, one simply calculates $\text{argMax}_y P(y \mid x)$ using the distribution P (U) represented by the Bayesian network. Thus

$$P(y \mid x) = P(U) / P(x)$$
$$= \prod_{u \in U} p(u \mid pa(u)) \quad (1)$$

where pa(u) is the set of parents of u in network structure.

From the observation of many studies, we found that evaluation of Bayesian network algorithm implementation usually results in high recall [17], hence it is chosen as one of the classifiers to be used in our approach.

4) J48 Decision tree Method

J48 Decision tree [18] is used to classify the given instances by constructing a decision tree using the training set data. J48 implements Quinlan's C4.5 algorithm for generating a pruned or expand C4.5 decision tree. The decision trees generated by J48 can be used for classification. J48 builds decision trees from a set of labeled training data using the concept of information entropy. J48 examines the normalized information gain (difference in entropy) that results from choosing an attribute for splitting the data and constructs the tree taking each attribute one by one. J48 can handle both continuous and discrete attributes, and also data with missing attribute values. Further it provides an option for pruning trees after creation.

J48 Decision tree algorithm performs feature selection as a part of the classification procedure. As this algorithm includes each attribute by analyzing its information gain, the attributes considered in this method forms a better feature set for classification. This algorithm is used in our method for performing the initial process of feature selection.

5) Combined Regression and Ranking(CRR)

CRR is a method that optimizes regression and ranking objectives simultaneously [8]. This method is applied to range of large-scale tasks, including click prediction for online advertisements. This combination guards against learning degenerate models that perform well on one set of metrics but poorly on another. Since we face a similar problem in our analysis, we chose to use this method. Another importance of this method is that it applied to the case of rare events or skewed sample distributions to improve regression performance due to the addition of informative ranking constraints. Since we are also facing imbalance problem in our sample distribution, CRR is employed as one of the method in this study.

6) Voting

Voting classifier is used for combining classifiers using un-weighted average of probability estimates (classification) or numeric predictions (regression) [12]. The algorithm takes an inducer and a training set as input and runs the inducer multiple times by changing the distribution of training set instances. The generated classifiers are then combined to create a final classifier that is used to classify the test set. This system also selects a classifier from the set of classifiers by

minimizing error on the training data. Voting method is a suitable method for the datasets suffering from the problem of class imbalance. Thus it is chosen to be experimented in our approach.

7) Cost Sensitive Stacking

Cost Sensitive Classification [13] extends the usual classification methods by coupling a cost vector c_i to each training sample (x_i, y_i). It introduces a methodology for extending regular classification algorithms to cost-sensitive ones with any cost. The j^{th} component c_{ij} denotes the cost to be paid when the label y_{ij} is misclassified. For a two class classification, the following matrix denotes the cost value:

	Actual Negative	Actual Positive
Predict Negative	0	C2
Predict Positive	C1	0

The entries [1, 2] (C1) and [2, 1] (C2) denotes the cost for misclassified labels. Thus this method tries its best to reduce the misclassifications. Cost sensitive stacking with a cost value associated with it denotes the cost of the misclassification on the actual positive column, but which is predicted negative. In our case, it denotes the cost value on the customers who actually belong to the class of accepting the campaigns but predicted to be the other. For example, cost 20 refers to the cost given to customers who were misclassified as those belonging to the class of not accepting the campaigns. The higher, the value given to this cost, the classifier will try its best to reduce this number of misclassifications.

Stacking is a method of combining the outputs of multiple independent classifiers for classification. Stacking uses the outputs of all classifiers, $f_1(x)$, $f_2(x)$, $..f_K(x)$ as features to form a new feature set $z = (z_1, z_2,...,z_K)$. Then, the new feature set together with the true label is used to learn the parameters w_{ij} of the stacking classifiers:

$$H_i(z) = \sum_{j=1}^{K} w_{ij} z_j , \quad (2)$$

where the weight w_{ij} will be positive if tag j is positively correlated to tag i; otherwise, w_{ij} will be negative. The stacking classifiers can recover misclassified tags by using the correlation information captured in the weight w_{ij}. This method boosts the results by reducing the misclassifications and so it is chosen as one of the boosting model for our experiments.

III. SYSTEM OVERVIEW

Fig 1 shows the block diagram representation of the proposed system. The customer details, obtained from the CDR (Call Detail Record) files are pre processed to remove the noise and outliers. Some of the attributes which are not relevant for campaign management process are also removed during pre-processing.

Generally CDR data consists of a number of related attributes. These attributes are considered as the simple or primary attributes. Some of the primary attributes

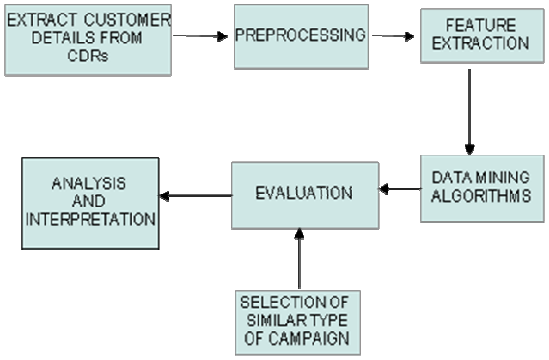


Fig 1. Block Diagram of the System

considered in our work are related to revenue direction technology used by the customer, plan, annual revenue etc. Some of the attributes are generated from the simple attributes which are known as *Derived Attributes*. These derived attributes gives a clear representation of the user behavior. Some of the derived attributes are average duration of the calls for each customer, total cost spent by the customer etc. Average duration of the calls is derived from the primary attributes like duration of the calls and the total number of calls. Similarly total cost spent by the customer is obtained by summing up the cost of all the calls made by the customer. The derived attributes along with the simple attributes together constitutes the input file for preliminary analysis. Those CDRs are aggregated for each customer and also on a weekly basis.

Feature extraction is the process of retrieving the useful attributes (features) from the input. Such features play an important role in contributing to the behavior of the customer in accepting the campaigns. Using J48 decision tree classifier as an embedded method of feature extraction gives a feature set of attributes that can be used for prediction. OneR rule classifier acts as a wrapper method of feature extraction to find out the single contributing feature for the campaign.

Other classification algorithms are used to predict the customer behavior in responding to the campaigns. Among the seven single classifiers and hybrid approaches used in this method, finally the classifier which performs well (i.e. which gives high precision, recall, F-measure etc) is selected and it is used to predict the behavior of the customers. The customers who were predicted to be the takers of the campaign are selected for launching new relevant campaigns.

A variety of campaigns exists in telecom domain. Each and every campaign will have its own contributing features. Some familiar types are bonus campaign, discount campaign etc. For each type of campaign, a separate model is framed and the process is performed individually.

For the evaluation of our study, we split the entire data available into training test which constitute the user behavior for a particular period and test set constituting the user behavior for a consecutive different period of time. The class label of the training and test set denotes the response of the customer to that particular campaign. Model is built by the classification algorithms using the training set and it is evaluated over the test set. Also a validation set consisting of the

training and test set together is constructed and a ten-fold cross validation is done on the set and results are analyzed to find out the usage of different classifiers and its predictions.

IV. EVALUATION MEASURES

The training and test data contains the details of those taking up the campaigns (class 0) and others (class 1). The work of a classifier is to construct a model by learning the training set and then predict whether a customer in the test set will belong to class 0 or class 1. Quality of the predictions is measured using the following matrix parameters.

	Predicted class 0	Predicted class 1
Actual class 0	True Positives (tp)	False Negatives (fn)
Actual class 1	False Positives (fp)	True Negatives (tn)

A. Precision

Precision is the degree to which repeated measurements under unchanged conditions show the same results. In other words, Precision is the fraction of retrieved instances that are relevant [9].

$$\text{Precision} = \frac{tp}{tp + fp} \quad (3)$$

B. Recall

Recall is the fraction of the documents that are relevant to the query that are successfully retrieved [9].

$$\text{Recall} = \frac{tp}{tp + fn} \quad (4)$$

C. F-Measure

F-Measure is the weighted harmonic mean of Precision and Recall [9].

$$\text{F-Measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

D. Accuracy

Accuracy is the degree of closeness of measurements of a quantity to that quantity's actual (true) value.

$$\text{Accuracy} = \frac{tp + tn}{tp + fp + tn + fn} \quad (6)$$

E. Kappa Statistic

Kappa Statistic can be defined as a measure of the degree of non-random agreement between observers or measurements of the same categorical variable and it is commonly used for the purpose of finding the agreement between the observers [19]. One of the uses of kappa is quantifying the actual levels of agreement. Kappa's calculation uses a term called the proportion of chance (or expected) agreement. This is interpreted as the proportion of times, the raters would agree by chance alone.

F. Area Under the Curve (AUC)

It is a measure to find the goodness of a classification algorithm by plotting a certain curve called the ROC curve and measuring the area under this curve [9].

G. Cross Validation

Cross-validation [20] is a technique for assessing how the results of a statistical analysis will generalize to an independent data set. One round of cross-validation involves partitioning a sample of data into complementary subsets, performing the analysis on one subset (called the *training set*), and validating the analysis on the other subset (called the *testing set*). In k-fold cross-validation, the original sample is randomly partitioned into k subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining k – 1 sub-samples are used as training data. The cross-validation process is then repeated k times (the folds), and the k results from the folds are averaged (or otherwise combined) to produce a single estimation.

V. EXPERIMENTAL RESULTS

Various CDR files containing the spent details, refill details and customer details for six months were combined together using customer mobile number as the primary key. This input CDR file contained approximately 1 million records. From this CDR file, training data is generated by taking the data for a period of 15 days and test data is generated for a period of 15 days consecutively. From these sets, aggregation was made for combining the details of each customer (using mobile number),and also on a weekly basis. After this aggregation, the training set contained 9463 records with 430 campaign acceptors. The test set contained 5581 records with 257 campaign acceptors were selected for running different experiments. Also a validation set combining the training and test set was generated. This CDR file with 39 attributes was pre processed to remove the missing values, outliers, error values and some of the attributes which was not necessary for our analysis.

From the pre processed CDRs, 27 primary attributes were selected. Out of the 27 attributes, 17 attributes were considered as such without any changes. 7 attributes were aggregated on a weekly basis for each unique customer. The remaining 3 attributes were used for generating 8 derived attributes. Thus our final data file consists of 32 attributes, 17 primary and 15 (7 + 8) derived and aggregated attributes. There is no restriction for the number of attribute usage.

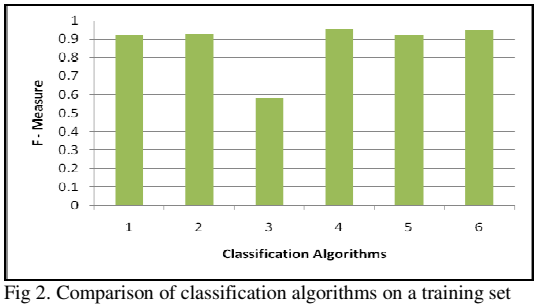


Fig 2. Comparison of classification algorithms on a training set

From another CDR which contained details about the campaigns such as campaign name, launch date and the customers who accepted the campaigns were retrieved. The available details also include the type of

campaign and the date of launch. From this data, information about every unique campaign was retrieved for our analysis. In this study, a discount campaign is considered for further analysis.

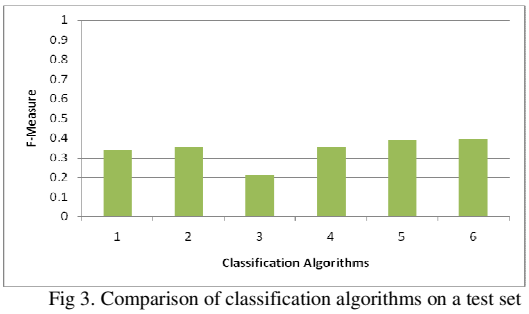


Fig 3. Comparison of classification algorithms on a test set

A. *Evaluation of Single Classifiers*

The classification algorithms are numbered as follows for easy representation: 1. OneR, 2. Logistic Regression, 3. Bayes Net, 4. J48 Tree induction, 5.Voting, 6. Cost Sensitive Stacking and 7. CRR. The evaluation measure (F-Measure) on the training set using 6 different classification algorithms is represented in Fig 2 and those on the test set are represented in Fig 3. There is comparatively a similar performance between classification algorithms on both training and test sets. Only Classification models 5 and 6 shows some improved performance on test set. It is clear that both models work on combining the results of different classifiers. Also AUC values generated on the test set, given in Fig 4 clearly illustrates the same that methods 5 and 6 outperforms the other single classifiers on the campaigning task.

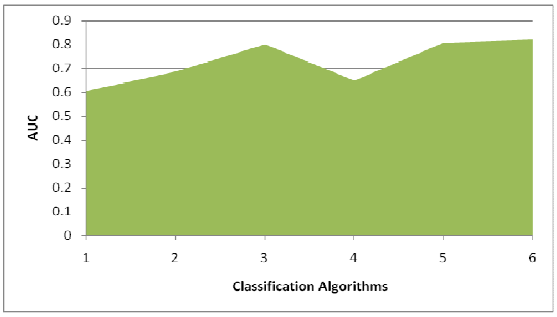


Fig 4. Comparison of classification algorithms – AUC

The results illustrated that a combination of the better performing individual classifiers provides better F-measure and AUC. Their combination was experimented by stacking them (Cost Sensitive Stacking) and through Voting. Moreover, Evaluation measures obtained on the test set are low when compared with that of the training set. It shows that the features obtained on the training set suits only up to 50% of the features on the test set. The results of the method CRR is given in Table 1, which clearly shows the vast improvement in recall value compare to other methods for different threshold levels, but precision value is very low. Our purpose is to get better F-measure for the campaign task without compromising the retrieval of relevant customers.

Table 1: Evaluation – Combined Regression and Ranking

Threshold	Precision	Recall	F-measure	AUC
1.0	0.085	0.676	0.151	0.713
0.9	0.076	0.962	0.141	
0.8	0.059	0.995	0.111	
0.7	0.053	0.995	0.099	

The evaluation measures on the validation set using 6 different classification algorithms on performing a cross-validation are tabulated in Table 2. The results demonstrate that the Precision, Accuracy and F- measure obtained on the cross validation sets are higher than that of the test set. This shows that the period of time considered between the training set and test set accounts for the changes in the features considered. Once the training set and test set are considered together, the features obtained are accurate and it results in satisfactory F-measure. As this is more suitable for a binary classification, Logistic Regression fits the function with least error and gives a high F-measure than other classifiers in this case.

Table 2: Evaluation measures on performing a cross validation

Method	Precision	Recall	F-Measure	Accuracy (%)	Kappa Statistic
1	0.776	0.467	0.583	96.804	0.5671
2	**0.871**	**0.469**	**0.61**	**97.1295**	**0.5963**
3	0.213	0.734	0.33	85.7599	0.2765
4	0.739	0.476	0.579	96.6915	0.563
5	0.746	0.465	0.573	96.6856	0.5569
6	0.508	0.641	0.567	95.3125	0.5423

Table 3. Evaluation of Hybrid Models on Test Set

Combination of Methods	Method performing well	Precision	Recall	F-Measure	Accuracy(%)	Kappa Statistic
1 and 2	**Cost Sensitive Stacking(Cost 20)**	**0.87**	**0.261**	**0.401**	**95.201**	**0.3838**
3 and 4	Cost Sensitive Stacking(Cost 10)	0.259	0.396	0.313	89.303	0.258
1 and 3	Voting-Simple Averaging	0.962	0.222	0.36	95.147	0.3453
2 and 3	Cost Sensitive Stacking (Cost 7)	0.613	0.283	0.387	98.477	0.3621
2 and 4	Cost Sensitive Stacking (Cost 15)	0.866	0.252	0.391	95.140	0.3731

B. *Evaluation of Hybrid Models*

Evaluation of the Hybrid approach is done by combining two of the single classifiers at a time and finding out the method (Voting, Cost Sensitive Stacking for various Costs) which gives good results. The results are tabulated in Table 3. The usage of hybrid model of combining OneR and logistic regression results better F-measure compare to other combinations.

VI. DISCUSSION

It is observed from Table 3 that the single classifiers which performed well (in terms of F-Measure) individually (Logistic and OneR) provided good F-measure when they were combined together using Cost Sensitive Stacking. The combination of the classifiers OneR and logistic regression, performing well shows that this campaign has one feature as an influencing attribute and here logistic classifier acts as a booster to improve the results. Usage of Cost 20 in Cost Sensitive Stacking proves that a higher cost minimizes the misclassifications on the specified class. Also the results obtained from CRR proves that combination of ranking with regression approach gives better results in terms of understanding suitable customers for campaigning in our model. So the methods can be implemented based on the requirements of the model.

In addition to performing multiple experiments, we have developed a GUI (Graphical User Interface) for automated campaign process which starts performing well for all the tasks from campaign generation to campaign tracking. A sample screenshot of the GUI is shown in Fig 5. Data mining techniques were useful in selection of features from the CDRs and thus to find out the proper target of customers to launch different types of campaigns. The automated provisioning of campaign management process became effective and economical in this process.

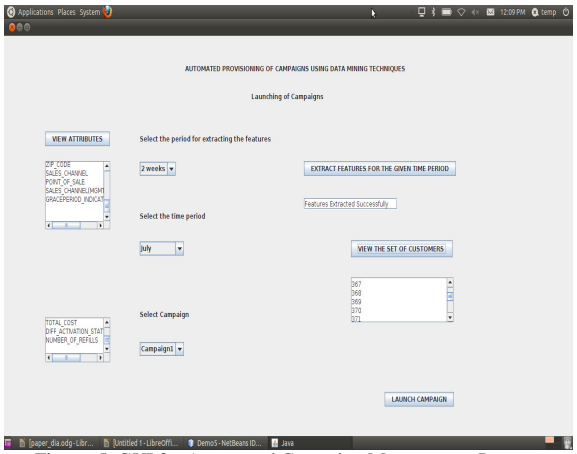


Figure 5. GUI for Automated Campaign Management Process

VII. CONCLUSION

Predicting the behavior of the customers relating to their campaign acceptance has been done successfully by extracting the important features. The experimental results are verified for a particular type of campaign. Similar method can be used for the other types and the groups of customers for which the campaign should be launched can be easily determined. The above result shows that an accuracy rate of 97.12 % and an F-measure of 40% can be reached in our method. Moreover, by the method of automotive provisioning, our approach becomes more innovative and effective in finding out the group of customers for launching the campaigns promptly.

ACKNOWLEDGMENT

This work was supported by Ericsson R&D, Chennai. We would like to thank my team and Research Manager, R& D Head, for moral support and encouragement to complete this work. Also we propose our sincere thanks to other colleagues Venkatesh and Vikas Verma for their excellent support in figure out specific models for our research activities.

REFERENCES

[1] http://www.businesslogicsystems.com
[2] http://www.xerox.com/downloads/usa/en/g/whitepapers/gdo_whitepaper_marketing_automation.pdf
[3] Michael J. A. Berry and Gordon S. Linoff, "Data Mining Techniques For Marketing, Sales, and Customer Relationship Management", Second Edition, Wiley Eastern, 2004
[4] Charles Elkan, "The Foundations of Cost Sensitive Learning," Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI), pp. 973-978, 2001.
[5] Buddhinath G, Derry D. A Simple Enhancement to One Rule Classification. Department of Computer Science & Software Engineering. University of Melbourne, Australia.
[6] Jie Cheng and Russell Greiner, "Learning Bayesian Belief Network Classifiers: Algorithms and System", Proceedings of 14th Biennial conference of the Canadian Society on Computational Studies of Intelligence: Advances in Artificial Intelligence, Springer- Verlag, London, UK, 2001
[7] Agresti A, "Building and applying logistic regression models". An Introduction to Categorical Data Analysis. Hoboken, New Jersey: Wiley. p. 138, 2007
[8] Schulley, D, "Combined Regression and Ranking", KDD'10, July 25-28, 2010, Washington, DC, USA.
[9] David MW Powers, "Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation", Technical Report SIE-07-001, extended version of paper presented in HC Science SummerFest, Dec 2007.
[10] Chu Chai Henry Chan, "Intelligent value- based customer segmentation method for campaign management: A case study of automobile retailer," Elsevier, pp.2754-2762,2008.
[11] Tarik Rashid, "Classification of Churn and non-Churn customers for Telecommunication Companies," International Journal of Biometrics and Bioinformatics (IJBB), Volume 3, Issue 5,pp. 82-89,2008.
[12] Yeshwanth V, Vimal Raj A, and Saravanan, M. "Evolutionary Churn Prediction in Mobile Networks using Hybrid Learning", in Proceedings of 24th International Florida Artificial Intelligence Research Society Conference (FLAIRS-24), May 18-20, 2011, Palm Beach, Florida, USA.
[13] Hung-Yi Lo, Ju-Chiang Wang, Hsin-Min Wang and Shou-De Lin, "Cost-Sensitive Stacking for Audio Tag Annotation an Retrieval", Proceeding of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp.2308-2311, 2011.
[14] Sandro Saitta, "Introduction to feature selection(part 1)," Data Mining Research, Sep 2007.
[15] Eugene Tuv and Alexander Borisov, "Feature Selection with Ensembles, Artificial Variables, and Redundancy Elimination," Journal of Machine Learning Research 10, pp.1341-1366,2009.
[16] Yvan Saey, Inaki Inza and Pedro Larranaga "A review of feature selection techniques in bioinformatics," Bioinformatics Advance Access, 2007.
[17] Edward O.Cannon, Ata Amini, "Support vector inductive logic programming outperforms the naive Bayes classifier and inductive logic programming for the classification of bioactive chemical compounds," Springer, Mar 2007.
[18] Ross Quinlan, "C4.5: Programs for Machine Learning,", Morgan Kaufmann Publishers, San Mateo, CA, 1993.
[19] Anthony J.Viera and Joannae M.Garrett, "Understanding Interobserver Agreement:The Kappa Statistic," Research Series, vol.37, no.5, May 2005.
[20] http://www.public.asu.edu/~ltang9/papers/ency-cross-validation.pdf

The impact of the observation of predictive features on the diagnosis of pigmented skin lesions and the therapeutic decision

Y. Wazaefi[1], A. Tenenhaus[2], A. Nkengne[3], J.F. Horn[4], A. Giron[4], S. Paris[1] and B. Fertil[1]
[1]LSIS, Aix-Marseille University, Marseille, France
[2] Department of Signal Processing and Electronic Systems, ESE Plateau de Moulon, Gif-sur-Yvette, France
[3]Johnson and Johnson Consumer, Skin Care Research Institute, Issy-les-moulineaux, France
[4]INSERM, Pierre et Marie Curie-Paris University, Paris, France

Abstract - *In this paper, we study the relationship between diagnosis and therapeutic decision on the one hand and the observations of the presence of ABCD features and some additional dermoscopic features of pigmented skin tumors on the other hand. The image database was composed of 227 images of pigmented skin lesions. Five senior dermatologists were asked for their expertise about these images. They gave their opinion about the presence of ABCD and dermoscopic features, their diagnosis and their therapeutic decision. The performances of dermatologists were evaluated in terms of their ability to diagnose melanoma by building statistical decision models from their observations of predictive features. Models allowed observing to what extent dermatologists ground their diagnosis on the malignancy features they detected. It appeared that a high variability of behavior among dermatologists is observed, concerning both the detection of features and the role of features for the elaboration of diagnosis.*

Keywords: ABCD features, melanoma diagnosis, decision model, Roc curve, Medicine Data Mining.

1 Introduction

As the survival rate of malignant melanoma depends on its thickness, diagnosis of malignant melanoma at an early stage could reduce the risk of mortality and increase the chance of prognosis considerably. The accuracy of the clinical diagnosis of melanoma with the unaided eye is only about 60%. Dermoscopy is a non-invasive in vivo technique for the microscopic examination of pigmented skin lesions, has the potential to improve the diagnostic accuracy [1]. Advances in objective dermatology diagnosis were obtained in 1994 with the introduction of the ABCD rule [2-3]. The ABCD rule specifies a list of visual features associated to malignant lesions (Asymmetry, Border irregularity, Color irregularity and Differential structure, i.e. size and number of structural features), from which a score is computed [4]. This methodology provided clinicians with a useful quantitative criterion, but it did not prove efficient enough for clinically doubtful lesions (CDL) essentially because ABCD features are difficult to characterize in those situations [5].

According to dermatologists' 'rules of good clinical practice', the diagnosis and associated therapeutic decision for black skin tumors is a multi-step procedure. The first step consists in detecting malignancy features (ABCD rule, 7-points checklist [6], etc.). In the second step, dermatologists combine these features according to their capacity in predicting malignancy. Stolz et al. has formulated a mathematical implementation of the ABCD rule [4]. Given that feature A may get a score varying from 0 to 2, feature B a score varying from 0 to 8, feature C a score varying from 1 to 6 and feature D a score varying from 1 to 5, a decision score (TDS) may be obtained by a linear combination of the features.

$$TDS = [(Asymmetry * 1.3) + (Border * 0.1) + (Color * 0.5) + (Differential\ Structures * 0.1)] \quad (1)$$

Tumors being given a TDS higher than 5.45 are considered highly suggestive of melanoma, an excision is recommended for tumors with a TDS higher than 4.8.

In order to build dermatologists' models of diagnosis/therapeutic decision, five senior dermatologists were asked to give their diagnosis and therapeutic decision for 227 images of tumors, together with their opinion about the existence of malignancy features (presence/absence). 'Models' of dermatologists were subsequently built by connecting predicted features to the so-called "gold standard" diagnostic (see below).

2 Materials and methods

The initial dataset used in this study was collected at the dermatology departments of the British Hertfort Hospital and the Louis Mourier Hospital in 'Ile de France' (France). A total of 900 images of pigmented skin lesions were

acquired in 'uncontrolled' conditions (see [7]). As a consequence of the inclusion protocol, many tumors were quite similar, and melanomas were largely in a minority. The current working database that initially included all identified melanoma lesions has been completed to 227 with randomly selected tumors. On doing so, it appeared that 77 lesions were classified as benign lesions. In order not to cause any needless distress to the patient, the majority of benign lesions were not surgically excised. Dysplastic lesions (i.e. atypical lesions, for which malignity may be suspected) were 118 in the database. Thirty-two pigmented lesions were categorized as malignant melanomas. The malignant melanomas and the dysplastic lesions were all surgically excised and histopathologically analyzed.

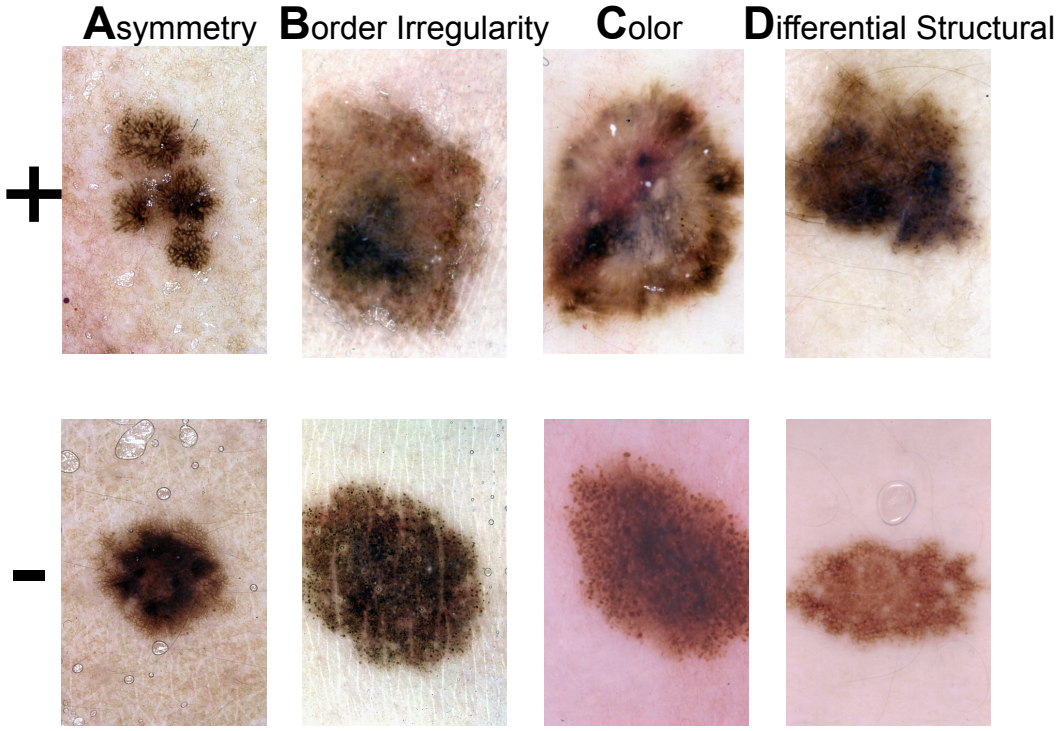


Fig. 1. Four nevi that fulfill ABCD rule (+) and four others that do not (-).

For this study, two classes were finally considered: histologically confirmed melanomas on the one hand and the remaining lesions on the other. For simplicity, this classification is referred to as the 'gold standard' diagnosis in this study.

Five senior dermatologists were asked for their expertise about the 227 selected images. They were presented each tumor both as macroscopic image and dermoscopic image. They subsequently gave their opinion about the presence of ABCD and dermoscopic features (dichotomic answers), their diagnosis (melanoma, dysplastic or benign lesion) and their therapeutic decision (dichotomic answer, excision/non-excision). Mimicking the Stolz's linear decision model, a logistic regression classifier [8] was built for each dermatologist using the features they reported as input and the 'gold standard' diagnosis as output, while a leave-one-out cross-validation was employed. The classifiers provide a probability to be a melanoma for each tested lesion in the selected database. ROC curves were built from these probabilities. They allows further analyzing the whole set of sensitivity/specificity couples of parameters. The area under the ROC curves (AUC) is a measure of the quality of prediction.

3 Results

As far as the diagnosis is concerned, one may observe a high variability of sensitivity among dermatologists whereas specificity remains similar, with the exception of the one obtained by dermatologist 3 (Table I).

TABLE I
Dermatologists' performances

Diagnosis and therapeutic decision	**Diagnosis** Sensitivity/Specificity	**Therapeutic decision** Sensitivity/Specificity
Dermatologist 1	0.62 / 0.90	0.84 / 0.63
Dermatologist 2	0.78 / 0.85	0.93 / 0.63
Dermatologist 3	0.59 / 0.71	0.84 / 0.39
Dermatologist 4	0.81 / 0.90	0.84 / 0.55
Dermatologist 5	0.71 / 0.80	0.87 / 0.63

Sensitivity and specificity are calculated with respect to the 'gold standard' diagnosis of melanoma.

In fact, the analysis of dermatologists' performances requires considering several factors. Sensitivity and specificity express the efficiency of the clinicians, but also the trade-off they believe to be acceptable with respect to the risk for a false diagnosis (Table I).

Depending on their level of confidence, they may privilege sensitivity over specificity. The opposite may also be true since it is a "risk-free" trial. The prior frequency of melanoma they meet usually in their daily practice may also play a role.

All these factors also take part in the therapeutic decision, although a much smaller one. In fact, we can expect (and observe) the therapeutic decision to have a higher sensitivity (as far as the prediction of melanoma is concerned), together with a lower specificity, since the CDL worthy of an excision encompasses melanoma. At the therapeutic decision level, sensitivities are more comparable, most melanomas are detected, but the cost (specificity) highly varies from one dermatologist to another.

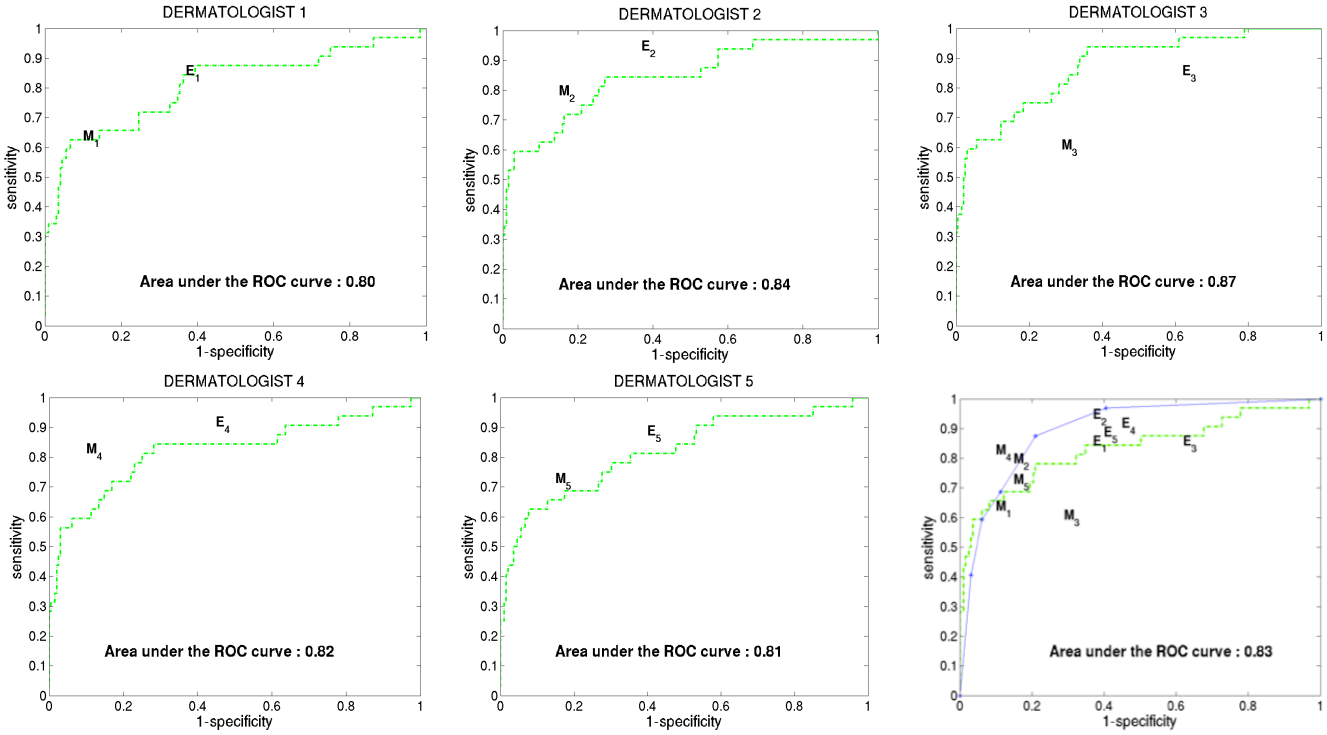


Fig. 2. Roc Curves for melanoma diagnosis result from logistic regression based on the features detected by each dermatologist. M_i and E_i show the accuracy of dermatologist i' diagnosis and therapeutic decision (Panels 1 to 5).
The last panel (bottom right) shows the Roc Curve of the logistic regression based on the consensual detected features (dotted line), together with the diagnosis and the therapeutic decision of each dermatologist. The 5-point solid line results from the voting schema about diagnosis so that the lower point corresponds to the tumors reported as melanoma by each of the 5 dermatologists, the next point corresponds to the tumors reported as melanoma by 4 out of the 5 dermatologists and so on.

Dermatologist' performances are shown, one at a time, in the subplots of Fig. 2. Sensitivity and specificity are displayed together with a ROC curve obtained with the mentioned linear classifier. It can be seen that dermatologist 1 grounds its diagnosis on the mere basis of the features he detected. Dermatologists 2, 4 and 5 probably use of additional visual features not available to the classifier, which makes their diagnosis and therapeutic decisions better than the results obtained by the classifier. Finally, dermatologist 3 seems poorly combining the features he has however efficiently detected. The best classifier performance is obtained from the set of features detected by the dermatologist 3, as shown by the AUC, which is the highest in this study.

Combining dermatologists' diagnoses and features characterization allows evaluation of the efficiency of the group of experts together. As dermatologists do not necessarily agree about the presence of features, diagnosis and therapeutic decision, a voting schema has been implemented (see reference [7] for details). It showed that full agreement between dermatologists is high (60%) as far as diagnosis is concerned, whereas therapeutic decision is more disputed (36%) (Table II). The picture is contrasted for the features: The agreement is high for asymmetry and relatively poor for color irregularity (Table II).

TABLE II
Distribution of the 227 images for ABCD features as a function of the dermatologists' vote

Feature	0-5	1-4	2-3
Asymmetry	54%	26%	20%
Border	36%	38%	26%
Color	34%	38%	28%
Differential structure	46%	30%	24%
Diagnosis	60%	23%	17%
Excision	36%	36%	28%

0-5 indicates that the 5 dermatologists are in full agreement, 1-4 indicates that 1 out of the 5 dermatologists disagrees and so on.

Combining diagnosis provides a remarkable result (Fig. 2, last panel, highest point of the 5-point solid line): 31 out of 32 melanomas are detected (sensitivity = 0.97) while cost remains low (specificity = 0.60). In contrast, the "consensual" ROC curve (AUC = 0.83) provided by the logistic model based on the consensual detected features does not reach the best available performance (0.87, Fig 2). Finally, the Stotz's formula, lightly adapted to fit our protocol, get an AUC of 0.79, which is quite good in this context.

4 Conclusion

In this study, five senior dermatologists were asked for their expertise about the 227 selected images. Models of diagnosis and therapeutic decision based on the observations of the presence of ABCD and dermoscopic features have been presented and evaluated. The results obtained show that the variability of performance of dermatologists is high, dermatologists with a melanoma-specific hospital activity showing the best performance, both for the diagnosis and the therapeutic decision.

The sensitivity and the specificity for diagnosis as well as therapeutic decision are higher if clinicians' advices are pooled. Such a result was not always assured, given the false positives to be cumulated.

Models also allow observing to what extent dermatologists ground their diagnosis on the malignancy features they detected. We believe that the clinical experience (based on the learning by sample paradigm) they gain during their daily practice is the key to their success.

5 Acknowledgment

This study was supported by the project I&M at LSIS, a grant "SKINAN" from ANR and a funding by VISOON

6 References

[1] H. Kittler, H. Pehamberger, K. Wolff, M. Binder, "Diagnostic accuracy of dermoscopy", *Lancet Oncol*, Vol. 3, pp. 159– 165, 2002.

[2] F. Nachbar, W. Stolz, T. Merkle, A. Cognetta, T. Vogt, M. Landthaler, P. Bilek, O. Braun-Falco, G. Plewig, "The ABCD rule of dermatoscopy: high prospective value in the diagnosis of doubtful melanocytic skin lesions", *Journal of the American Academy of Dermatology*, Vol. 30(4), pp. 551–559, 1994.

[3] R. Feldmann, C. Fellenz, F. Gschnait. "The ABCD rule in dermatoscopy: analysis of 500 melanocytic lesions", *Hautarzt*, Vol. 49, pp. 473–76, 1998.

[4] W. Stolz, A. Riemann, B. Armand, et al. "ABCD rule of dermatoscopy: a new practical method for early recognition of melanoma", *Eur J Dermatol*, Vol. 4, pp. 521–27, 1994.

[5] G. Capdehourat, A. Corez, A. Bazzano, P. Musé, "Pigmented skin lesions classification using dermatoscopic images", *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, Vol. 5856, pp. 537-544, 2009.

[6] R.H. Johr, "Dermoscopy: alternative melanocytic algorithms? The ABCD rule of dermatoscopy, menzies scoring method, and 7-point checklist", *Clinics in Dermatology*, Vol. 20(3), pp. 240–247, 2002.

[7] A. Tenenhaus, A. Nkengne, J.F. Horn, C. Serruys, A. Giron, B. Fertil, Detection of melanoma from dermoscopic images of naevi acquired under uncontrolled conditions, *Skin Research and Technology*, Vol. 16(1), pp. 85-97, 2010.

[8] J.G. Liao, Khew-Voon Chin. Logistic regression for disease classification using microarray data: model selection in a large p and small n case, *Bioinformatics*, Vol. 23 (15), pp. 1945-1951, 2007.

Conceptualization of Sentence Paraphrase Recognition with Semantic Role Labels

R. Yadav[1], A. Kumar[1], A. Vinay Kumar[2], and P. Kumar[1]
[1]Information Technology & Systems, Indian Institute of Management, Lucknow, Uttar Pradesh, India
[2]Finance & Accounting, , Indian Institute of Management, Lucknow, Uttar Pradesh, India

Abstract - *Sentence paraphrase recognition plays an important role in many NLP applications. In majority of previous studies, basic unit of information for analysis is the sentence itself. However, a sentence contains information about one or more events/entities in its multi-clausal structure. Clauses – conceptualized as Semantic Role Labels (SRL) or predicate-argument tuples – are the smallest grammatical units with which sentence information can be comprehended and compared with. Objective of this paper is to propose a sentence paraphrase recognition methodology using predicate-argument tuples as the basic unit of information.*

This paper introduces concept of paraphrasing, loosely paired, and unpaired tuples to establish sentence-sentence similarity. Two sentences are paraphrasing if they contain at least one paraphrasing tuple; and no or insignificant dissimilarities (loosely paired and unpaired tuples). The paper proposes two tuple representation schemes – first Vector Space Model based, and second based on distributed word representations (embeddings) learnt using deep Neural Network language models.

Keywords: Paraphrase Recognition, Semantic Role Labels, Predicate argument, Vector Space Model, Recursive Auto-Encoders

1 Introduction

Any two natural language expressions are called paraphrase (*para-* 'expressing modification' + *phrazein* 'tell') if both convey similar information or meaning. Sentence paraphrase recognition is essentially a *boolean* sentence-sentence similarity metric that is indispensable to many Natural Language Processing (NLP) applications like question-answering [1], text summarization [2], machine translation [3] etc. Understanding a sentence – with all its possible deep syntactic structure variations, language semantic nuisances like synonyms, idioms etc. – has been a challenging task. In majority of previous studies ([4], [5], [6], [7]), basic unit of analysis for sentence comprehension is the sentence itself. However, a sentence is generally a multi-clause grammatical structure – referred to as S*emantic Role Labels* (SRL) or *predicate-argument tuples* – conveying information about more than one event/entity at a time. SRLs or predicate-argument *tuples* are the smallest grammatical unit of information that can be used to comprehend sentence meaning [8]. Objective of this study is to propose a sentence paraphrase recognition methodology with SRLs as its basic unit of analysis.

Defined for each instance of sentence's predicate (verb), semantic role labeling entails assigning role of WHO did WHAT to WHOM, WHEN, WHERE, WHY, HOW etc. according to predicate's verb frame [9] – collectively called predicate and its corresponding arguments. This paper uses the term predicate-argument *tuple* or just *tuple* interchangeably to refer to predicate and its argument SRLs. A sentence *S* having *m* predicates is set of *m* predicate-argument *tuples*:

$$S = \{PA_1, PA_2, \dots, PA_m\} \text{ with } m \geq 1 \qquad (1)$$

where, $PA_i = \{p_i, a_{i0}, a_{i1}, \dots a_{iK}\}$ with $1 \leq i \leq m$

Here, *K* is the size of domain of arguments labeled by a Semantic Role Labeler ([8], [9]).

For instance, sentence example S_{EX1} – *"Amrozi accused his brother, whom he called "the only witness", of distorting his evidence."* – has three predicates – *accused, called,* and *distorting.*

Amrozi$_{[p1a0]}$ *accused*$_{[p1]}$ *(his brother)*$_{[p1a1,\ p2a1,\ p3a1]}$ *...* *(whom*$_{[p2ar1]}$ *he*$_{[p2a0]}$ *called*$_{[p2]}$ *"(the only witness)*$_{[p2a1]}$*")*$_{[p1a1]}$ *...(of distorting*$_{[p3]}$ *(his evidence)*$_{[p3a2]}$*)*$_{[p1a2]}$

Hence, sentence *S* can be represented as set of three predicate argument *tuples* (*m = 3*) as depicted in Table I.

TABLE I
UNDERSTANDING SENTENCE S_{EX1} WITH PREDICATE-ARGUMENT *TUPLES*

T.I.	**p**	**a_0**	**a_1**	**a_2**
PA_1	*accused*	*Amrozi*	*his brother*	*of distorting his evidence*
PA_2	*called*	*he*	*whom, the only witness*	
PA_3	*distorting*	*his brother*	*his evidence*	

In a well written sentence, *m* will generally be less than five. Microsoft Research (MSR) Paraphrase Recognition Corpus [10] has on an average m = 2.24 [11]. For comparing two sentences S_1 and S_2, each *tuple* of sentence needs to be compared with each *tuple* of another sentence.

From all the possible pairings $S_1 X S_2$, Qiu et al [11] defined two categories of *tuples* – semantically paired *tuples* and unpaired *tuples*. This paper extends it to following three categories of possible pairings:

1. *Semantically paired tuples* or *Paraphrasing tuples* – These *tuples* convey similar meaning about same event or same entities of a sentence pair.
2. *Loosely paired tuples* – These *tuple* pairs are responsible for conveying same part of information content of each sentence. They may talk about same event or same actors, but they do not convey same meaning. Loosely paired *tuples* help in identifying unpaired *tuples* that have no counter-part in the other sentence.
3. *Unpaired tuples* – *Tuples* that are neither loosely paired nor semantically paired are unpaired *tuples* of a sentence pair.

It is assumed that the given sentences are from same context [12]. Concept of pairing is elucidated below with the help of three hypothetical sentences:

S_1: *Amrozi accused his brother, whom he called "the only witness", of distorting his evidence.*
S_2: *Amrozi accused his brother of deliberately altering his evidence.*
S_3: *Referring to him as a liar, Amrozi accused his brother of deliberately distorting his evidence.*

S_1 has three predicates = {*accused*, *called*, *distorting*}; *S_2* has two predicates = {*accused*, *altering*}; *S_3* has three predicates = {*Referring*, *accused*, *distorting*} as shown in Table II.

TABLE II
COMPARING SENTENCES S_1, S_2, S_3 WITH PREDICATE-ARGUMENT TUPLES

	T.I.	p	a0	a1	a2
S_1	PA_{11}	*accused*	*Amrozi*	*his brother*	*of distorting his evidence*
	PA_{12}	*called*	*He*	*whom, the only witness*	
	PA_{13}	*distorting*	*his brother*	*his evidence*	
S_2	PA_{21}	*accused*	*Amrozi*	*his brother*	*of deliberately distorting his evidence*
	PA_{22}	*altering*	*his brother*	*his evidence*	
S_3	PA_{31}	*referring*	*Amrozi*	*him as a liar*	
	PA_{32}	*accused*	*Amrozi*	*his brother*	*of deliberately distorting his evidence*
	PA_{33}	*distorting*	*his brother*	*his evidence*	

Comparing sentences *S_1* and *S_2*, (*PA_{11}*, *PA_{21}*) and (*PA_{13}*, *PA_{22}*) are semantically paired while *PA_{12}* is unpaired and insignificant to its sentence meaning – hence *S_1* and *S_2* are paraphrase. Comparing sentences *S_1* and *S_3*, *(PA_{11}, PA_{32})* and *(PA_{13}, PA_{33})* are semantically paired while *(PA_{12}, PA_{31})* are loosely paired and significant to the meaning of two sentences – hence *S_1* and *S_3* are not paraphrase. Hence, a sentence pair is paraphrasing if it contains semantically paired *tuples* and has no or insignificant dissimilarities [11]. Sentence-sentence similarity can be established in terms of semantically paired or paraphrasing *tuples* while dissimilarities in terms of loosely paired and unpaired *tuples*.

Qiu et al [11], in their work on paraphrase recognition using SRLs, decomposes sentence paraphrase recognition task into two predicate-argument *tuple* level tasks – first is semantically paired *tuples* identification heuristic and second is unpaired *tuple* significance classification. Authors represented SRLs with their respective syntactic headword [9] and compared them using Lin's thesaurus similarity metric [13]. Similarity between two *tuples* is established using a weighted average of similarity between their predicate and argument labels. Sentence pair similarities and dissimilarities are identified heuristically in terms of semantically paired *tuples* and unpaired *tuples* respectively. Two sentences are paraphrasing if any dissimilarity (unpaired *tuples*) present is insignificant. Qiu et al [11] derives *tuple* significance training data set from MSR paraphrase recognition data set [10] to learn dissimilarity significance classification. Authors [11] reported recall of 0.934 and precision of 0.725. Low precision is attributed mainly to paraphrase recognition failure in case of complex multi clause sentences [11]. This could be primarily because authors approximate SRL phrases with their syntactic headword which risk losing significant information in case of long SRL phrases.

This paper proposes two predicate-argument *tuple* representation schemes – first, Vector Space Model (VSM) based representation; and second, deep Neural Network language model based word and phrase embeddings ([14], [15], [16]). For comparing two *tuples*, *tuple* paraphrase recognition is learnt as a separate classification task with *tuple-tuple* similarity matrix as the feature set. Like Qiu's work [11], this paper formalizes sentence level paraphrase recognition into two *tuple* level classification tasks – first, *tuple* level paraphrase recognition and second dissimilarity significance classification. In order to derive training data sets for these two *tuple* based classification tasks from sentence based MSR paraphrase corpus [10], concept of loosely paired *tuples* is introduced. Loosely paired *tuples* discuss same event or same entity in a sentence pair, but are not semantically similar or paraphrasing. This category of *tuple* pairs helps in deriving negative examples for *tuple* paraphrase recognition data set and also helps in refining and enhancing unpaired *tuple* significance classification data set.

The paper presents an SRL based paraphrase recognition approach in following sections. Section 2 gives an overview of SRL based representation schemes adopted in past and characteristics of an efficient representation scheme needed for paraphrase recognition task. It further proposes two SRL representation schemes - Vector Space Model (VSM) based representation; and second, deep Neural Network language model based word and phrase embeddings ([14], [15], [16]). Overall sentence paraphrase recognition methodology is

discussed in section 3. This is followed by conclusion and future directions in Section 4.

2 Semantic Role Labels Representation

To the best of our study and knowledge, Qiu's work has been the only work that deploys SRLs to represent and compare two sentences. Qiu et al [11] reduced each SRL phrase to its corresponding syntactic headword feature and used Lin's thesaurus based word-word similarity measure as θ.

Reducing an SRL phrase with its syntactic headword often lose the main phrase content or the words modifying meaning of the phrase content. For instance, for sentence S_{EX2}

S_{EX2}**:** *Revenue in the first quarter of the year dropped 15 percent from the same period a year earlier.*

Considering arguments for predicate "*drop*", syntactic head word[1] for a_0 noun phrase "*revenue in the first quarter of the year*" is reduced to "*year*" while syntactic head word for a_2 prepositional phrase "*from the same period a year earlier*" is reduced to "*period*" (see **Error! Reference source not found.**). It is observed that approximating a phrase with its syntactic headword may risk losing information significant in paraphrase recognition.

TABLE III
SYNTACTIC HEADWORD AND VSM BASED TUPLE REPRESENTATION SCHEMES FOR S_{EX2}

	Predicate (p)	**Arg0 (a_0)**	**Arg1 (a_1)**	**Arg2 (a_2)**
PA_1	*dropped*	*revenue in the first quarter of the year*	*15 percent*	*from the same period a year earlier*
Syntactic Headword	*dropped*	*year*	*percent*	*period*
VSM features	*drop*	*revenue, first, quarter, year*	*%NUMBER%, percent*	*same, period, year, earlier*

Qiu [11] used Lin's thesaurus based word-word similarity measure for comparing two phrases headwords. Lin thesaurus [13] major drawback is consideration of antonyms and unrelated words as proximate neighbours of a word. Further, Lin thesaurus [13] similarity measure between two words is independent of sentences' context or discourse and hence sentence context has no role to play in disambiguating accurate sense of a word.

Predicate-argument *tuple*, $\boldsymbol{PA} = \{p, a_0, a_1 \ldots a_K\}$ is an ordered collection of a sentence's predicate phrase (verb/event) and its corresponding argument label phrases. With number of argument types *K* fixed, *tuple* representation is essentially a function of a phrase representation scheme ψ such that, given an appropriate similarity metric θ, similar phrases have high similarity and dissimilar phrases have low dissimilarity.

In MSR paraphrase corpus [10], each sentence has on an average 2.25 *tuples*, and hence comparing two sentences needs on an average approximately five *tuple-tuple* comparisons. Hence, an efficient *tuple* or phrase representation scheme ψ, given an appropriate choice of θ, should facilitate a fast *tuple-tuple* comparison. Lexical string based phrase representations needs elaborate string to string comparison metric and hence cannot support a fast *tuple-tuple* comparisons. This paper suggests two vector based *tuple* representation schemes. First representation scheme ψ_{VSM} is based on Vector Space Model (VSM) with binary weights – signifying presence or absence of a feature. Second representation scheme ψ_{RAE} is based on deep Neural Network Language Model trained word and phrase embeddings ([14], [15], [16]).

2.1 Vector Space Model based SRL representation scheme

First representation scheme ψ_{VSM} is based on Vector Space Model (VSM) with binary weights – signifying presence or absence of a feature. Feature definition for ψ_{VSM} is lemmatized content words (noun, verb, adjective, adverbs). Features are normalized with numbers and Named Entity (person, location, percentage, currency, title, company) based abstractions (see **Error! Reference source not found.**). Pronouns are treated as wildcards for all possible named entities of the corresponding sentence pair. Feature vocabulary V_S is local to a given sentence pair, where each phrase of that sentence pair can be represented with a $|V_S|$ size binary vector. Suggested choices of similarity metric θ are cosine similarity metric or Jaccard similarity metric.

Since predicate phrase is one of the most important element of a *tuple* in paraphrase recognition [11] and verbs being one of the most polysemous in nature, it is important to incorporate a verb sense disambiguation algorithm for an improved recall. This paper implements a verb disambiguation algorithm based on Galley's [17] linear order lexical chain based noun disambiguation algorithm. Author [17] scans text to identify candidate words (nouns) while simultaneously creating a disambiguation graph where all words are attached with weighted edges with respect to following semantic relations – synonym, hyponym, hypernyms and coordinate words. This paper scans a sentence pair for candidate verbs (and possible verb nominalizations [18]) creating a graph where all verbs are attached with positive weighted edges with respect to following semantic relations – synonym, hypernyms, entailment and coordinate verbs; and with negative weighted edges for antonym relations. On testing the algorithm on around 200 MSR paraphrase sentence-pairs, it is able to detect verb relation between sentence-pair instances like "*...share were up...*", "*...shares jumped...*", "*...shares rose...*", or "*...shares increased...*" successfully. It is asserted that proposed verb

[1] Syntactic head of a phrase calculated using a head word table described in [22], Appendix A with modifications on prepositional phrases proposed by [9]

disambiguation algorithm ought to improve recall for sentence pair recognition task.

2.2 Recursive Auto-Encoder based SRL representation scheme

Another area of emerging interest in NLP is use of deep neural language models [19] to learn distributed word representations ([14], [15]) or phrase representations [16] in an unsupervised manner such that these models can be reused in other specific supervised NLP tasks like POS tagging [20], NER recognition [20], paraphrase recognition [16] etc.

Distributed word representation (or embeddings) is a d-dimensional vector such that semantically or syntactically similar words have embeddings closer to each other. This leads to a smoother solution space with lesser discontinuities and hence model trained on such a space will have better generalization on unseen data [14]. The word embedding matrix , where |V| is the size of vocabulary, is learnt jointly as part of an unsupervised deep neural language model ([14], [15]). Such unsupervised neural language models based on distributed word representation are reusable in other NLP tasks.

Using *Turian* embeddings [15] as word representation scheme, Socher et al [16] introduced auto-encoder – an unsupervised neural network model – that encode a bi-gram embedding of size *2*d* into a *d* size vector such that the bi-gram can be reconstructed back with minimum reconstruction error. To encode a sentence, this auto-encoder is applied recursively on sentence's syntactic parse tree in right-to-left bottom-up manner such that all its non-leaf node phrases are encoded into a fixed *d* size vector minimizing unfolding reconstruction error at each node – the model referred to as unfolding Recursive Auto-Encoders (RAE) [16]. Once a word embedding matrix *L* and an RAE is learnt on an unlabeled corpus, these can be re-used to encode any sentence's syntactic parse tree. Socher et al [16] applied unfolding RAE based sentence representation for paraphrase recognition reporting state-of-the-art accuracy of 76.8%.

This paper proposes to use RAE-encoded parse tree for predicate-argument *tuple* representation. However, a syntactic parser like Stanford parser [21] adheres to grammatical understanding of predicate while SRL literature [8] is based on logical understanding of predicate. From grammatical perspective, a sentence has two components – the subject; and the rest of the sentence part called predicate that modifies the subject. On the other hand, Proposition bank [8] – the annotated dataset used for SRL task – defines predicate as the verb and its related adverbs or auxiliaries modifying the verb; while arguments are the subjects, direct or indirect objects of the predicate defined as per corresponding verb frame [8]. Passing sentence syntactic parse tree to Socher's RAE will encode each intermediate node of the tree in a fixed size vector. However, phrases corresponding to predicates are not preserved with these nodes. This difference is best elucidated with following example sentence and its parse tree (Fig. 1. Syntactic Parse Tree for Sentence SFig. 1). In sentence S, predicate phrase is "*denied to accuse*" which is not preserved in any of its syntactic parse tree nodes.

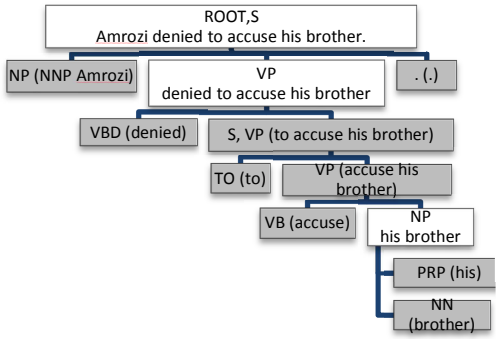


Fig. 1. Syntactic Parse Tree for Sentence S

S: "*Amrozi denied to accuse his brother.*"

Hence parse tree needs to be transformed at every verb phrase such that its logical predicate phrase and its arguments

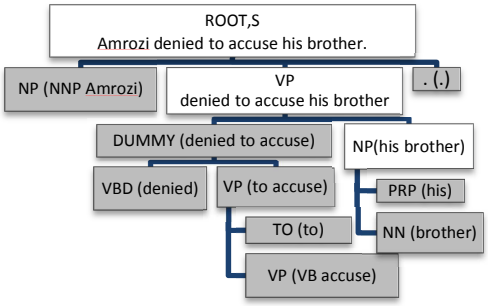


Fig. 2. Transformed syntactic parse tree so that predicate SRL phrase is preserved

are preserved in its intermediate nodes as shown in Fig. 2.

However, Socher' RAE is trained on Stanford parser's syntactic parse tree and one needs to verify whether the RAE gives same quality of encodings with transformed parse tree too i.e. with no significant increase in reconstruction error. This was verified on 200 sentence-pair sample taken from MSR paraphrase corpus. The change in reconstruction error of a sentence ROOT node encoding was observed to be insignificant with *p-value* 0.021. This verifies that Socher's RAE can be safely used with transformed trees preserving predicate phrases.

Fixed size encodings of the parse tree thus extracted are used for predicate-argument *tuple* representation. Hence each *tuple* can be encoded in *(K+1)Xd* size matrix where *K* is the number of argument SRLs and *d* is the size of word embeddings used in Socher's RAE. Suggested choice for comparing two phrases is Euclidean distance.

3 Methodology

In previous section, two SRL representation schemes ψ_{VSM} and ψ_{RAE} were proposed where each SRL is represented as $|V_S|$ size and d size vector respectively, where $|V_S|$ is the size of local sentence-pair's unique feature set while d is the size word embeddings. Choice of θ proposed for ψ_{VSM} is cosine similarity and Jaccard measure while choice of θ proposed for ψ_{RAE} is Euclidean distance metric. Let the choice of SRL representation scheme and similarity measure be ψ and θ in general.

3.1 Tuple-Tuple Similarity Matrix

Given two sentences S_1 and S_2 having m and n predicate-argument *tuples* respectively, and PA_{i1} and PA_{i2} are the i^{th} and j^{th} *tuple* from S_1 and S_2:

$$S_1 = \{PA_{11}, PA_{12}, \dots, PA_{1m}\} \text{ and} \quad (2)$$
$$S_2 = \{PA_{21}, PA_{22}, \dots, PA_{2n}\} \quad (3)$$

$$PA_{1i} = \{p_{1i}, a_{1i0}, a_{1i1}, \dots a_{1iK}\};\ PA_{2j} = \{p_{2j}, a_{2j0}, a_{2j1}, \dots a_{2jK}\} \quad (4)$$

Where, K is the size of domain of arguments labelled by Semantic Role Labeller. A predicate generally has two to four arguments [8] and hence maximum arguments will be *null*. Each SRL is encoded using SRL representation scheme ψ.

For comparing two sentences, one need to consider all possible *tuple* comparisons to find semantically paired, loosely paired and unpaired *tuples*. For comparing any two *tuples* PA_{1i} and PA_{2j}, this work proposes to use a similarity matrix similar to Socher's [16] work using similarity metric θ. Matrix is subsequently normalized such that each entry lies between zero and one. Unlike Socher's work, pooling is now already defined where each SRL element of a *tuple* is pooled into one region. Similarity of each pooled rectangular region is calculated using *max* operator (*min* operator in case of Euclidean distance metric). Resulting pooled (K+1)X(K+1) matrix can be fed to a classifier for learning paraphrasing characteristics.

Tuple-tuple similarity matrix thus created not only accounts for element-wise similarity but also captures cross SRL alignment between two *tuples*. For instance in following sentence pair:

S_1: *Troy is sentenced to life in prison without parole.*

S_2: *Troy face life sentence in prison without parole.*

$S_1 = \{PA_{11}\}$ and $S_2 = \{PA_{21}\}$

PA_{11} = {[*p*, "*sentenced*"], [a_0, *null*], [a_1, "*Troy*"], [a_2, "*to life in prison*"], ..., [a_{man}, "*without parole*"], ...}

PA_{21} = {[*p*, "*face*"], [a_0, "*Troy*"], [a_1, "*life sentence*"], ..., [a_{loc}, "*in prison*"], [a_{man}, "*without parol*"], ...}

Here, predicates for PA_{11} and PA_{21} are "*sentence*" and "*face*" respectively. Both these verbs follow different verb frames and hence argument a_1 of "*sentence*" is argument a_0 of "*face*", argument a_2 of "*sentence*" is argument *location* of "*face*", etc. Similarity matrix captures verb-frame alignment for these two verbs efficiently as shown in Table IV (with darker shade signifying higher similarity).

TABLE IV
SIMILARITY MATRIX BETWEEN *TUPLES* PA_{11} AND PA_{21}

without parol	a_{man}						
null	a_{loc}						
to life in prison	a_2						
Troy	a_1						
null	a_0						
sentenced	p						
		p	a_0	a_1	a_2	a_{loc}	a_{man}
		face	Troy	life sentence	null	in prison	without parol

Further, the predicate-argument *tuple* vectors represented with a predicate and its K argument type phrases ought to have majority of its elements *null* leading to a sparse *tuple-tuple* similarity matrix. Instances of comparison of a *null* SRL with non *null* and *null* SRLs need to be handled separately as both cases hold different information for *tuple-tuple* comparison.

3.2 Sentence Paraphrase Recognition Training

ψ and θ, sentence level paraphrase recognition task is divided into two phases – first, learning *tuple* level paraphrase recognition classification ξ_P and second learning dissimilarity significance classification ξ_D. Hence, sentence-pair paraphrase detection training data set needs to be translated to create training data set for paired *tuples* paraphrase recognition and dissimilarity significance classification tasks.

In phase I, once sentences are represented as set of their predicate argument *tuples* using ψ, unpaired *tuples* in a sentence pair are identified. This can be learnt by training a decision tree classifier on manually labelled sample of sentence pairs (around 200). In MSR paraphrase corpus, following sentence-pair types are relevant for first sub-task:

SP₁: Sentence pair where each sentence has only one predicate-argument *tuple*

SP₂: Sentence pair that is paraphrasing and has only one paired *tuple* (loosely or semantically) and only one unpaired *tuple*

Predicate-argument *tuple* pairs belonging to SP_1 and SP_2 sentence pairs make data-set for first task, where predicate-argument *tuple* pair is labelled similar if sentence pair is paraphrasing and dissimilar otherwise. Predicate-argument *tuple-tuple* similarity matrix is used as features for training *tuple* semantic similarity classifier ξ_P.

Using ξ_P, paraphrase training data set is tested for semantically paired and loosely paired *tuples*. For learning

dissimilarity significance in sentence pair paraphrasing, this paper follows Qiu's approach [11] except that authors defined dissimilarity in terms of unpaired *tuples* only while our work defines dissimilarity in terms of unpaired and loosely paired *tuples*. Following sentence-pair types are relevant for creation of dissimilarity significance classification data set:

UP$_1$: Sentence pair that is non-paraphrasing; and that has only one unpaired or only one loosely paired *tuple* and all paired *tuples* are semantically paired.

UP$_2$: Sentence pairs that is paraphrasing; and that has at least one semantically paired tuple

Unpaired *tuple* or loosely paired *tuple* in sentence pair belonging to UP_1 is significant, as despite of all paired *tuples* being similar the sentence pair is non-paraphrasing. Similarly, all unpaired *tuple* or loosely paired *tuples* in sentence pair of type UP_2 are insignificant as despite of its presence, the sentence pair is paraphrasing. Qiu et al [11] used *n*-gram ($n = 4$) syntactic path between predicate of unpaired *tuple* and paired *tuple* with the closest shared ancestor. Apart from four-gram syntactic path features, other features presumed to be of significance for the task are WordNet [18] verb category (after verb sense disambiguation as mentioned in VSM based representation scheme above), number of children *tuple's* predicate has in its sentence's transformed parse tree, and whether predicate has any arguments other than subject (*arg0*).

3.3 Sentence Paraphrase Recognition Testing

A sentence pair is paraphrasing if it has at least one semantically paired *tuples* and all its dissimilarities (loosely paired and unpaired *tuples*) are insignificant in paraphrasing or meaning.

Methodology can be summarised as follows:

Input:

Sentence Pair ***SS*** = $\{S_1, S_2\}$

Labelled training Data-set D_r with N_r sentence pairs $\{ S_{i1}, S_{i2}, p_i\}$ where p_i is 1 if ith sentence pair is paraphrases and 0 otherwise

Output: Predicted paraphrase status P (0/1) for SS

TRAINING BEGIN

Step 1: Represent each sentence in D_r as set of its predicate-argument *tuples* using ψ

Step 2: For each sentence pair in D_r, identify unpaired *tuples* and paired (loosely/semantically paired) *tuples* using heuristic K

Step 3: Select paired *tuples* of SP_1 and SP_2 type sentence pairs to form a training data-set D_r^p for training *tuple* paraphrase classifier ξ_P

Step 4: Create *tuple-tuple* similarity matrix as feature set to learn ξ_P

Step 5: Label each *tuple* pair in D_r^p as paraphrasing if corresponding sentence pair is paraphrasing and non-paraphrasing otherwise

Step 6: Train ξ_P

Step 7: For each sentence pair in D_r, identify paraphrasing *tuples* and loosely paired *tuples* using ξ_P

Step 8: Select unpaired and loosely paired *tuples* of UP_1 and UP_2 type sentence pairs to form a training data-set D_r^D for training dissimilarity significance classifier ξ_D

Step 9: Create following feature-set for learning ξ_D:

n-gram features of shortest parse tree path of *tuple's* predicate from predicate of any paired *tuples* of the sentence.

WordNet verb category

Number of children *tuple's* predicate has in its sentence's parse tree

Whether it has any arguments other than arg0

Step 10: Label each *tuple* pair in D_r^D as significant if corresponding sentence pair is of type $UP_1$and non-paraphrasing if it is of type UP_2

Step 11: Train ξ_D

TRAINING END

SS PARAPHRASE DETECTION BEGIN

Step 1: Represent each sentence in SS as set of its predicate-argument *tuples* using ψ

Step 2: Identify unpaired *tuples* and paired (loosely/semantically paired) *tuples* using heuristic K

Step 3: Identify paraphrasing *tuples* and loosely paired *tuples* using ξ_P using *tuple-tuple* similarity matrix. If no semantically paired *tuple* found, return P as 1 else go to step 4.

Step 4: Find significance of unpaired and loosely paired tuples using ξ_D

Step 5: If all unpaired and loosely paired tuples are insignificant return P as 1 else return 0.

SS PARAPHRASE DETECTION END

4 Conclusion and Future Directions

Semantic role labels or predicate-argument *tuples* are the smallest grammatical units using which a sentence meaning can be appropriately captured. Qiu et al [11] proposed sentence paraphrase recognition methodology using predicate-argument *tuples* as the basic unit if information. However, Qiu et al [11] approach lacks in efficient SRL representation methodology and relies on thesaurus based heuristic to identify paraphrasing *tuples*. This paper proposed two improvisations on Qiu et al [11] approach. First, two *tuple* representation schemes are proposed – VSM based and RAE based representations. The two vector based representation schemes are asserted to deliver faster and accurate SRL phrase comparison. Second, the paper introduced concept of loosely paired *tuples* in order to formalize *tuple* paraphrase recognition problem as a separate classification task. For comparing two *tuples*, use of *tuple-tuple* similarity metric is suggested as it efficiently captures SRL alignment corresponding to polysemous verb frames. Paper also proposes a verb sense disambiguation algorithm which has been validated manually on 200 sentence pairs from MSR paraphrase corpus.

The paper proves to be a blue-print for SRL based sentence paraphrase recognition. Further, one should verify which representation scheme best captures the paraphrasing features of a sentence pair. Also, appropriate handling of *null* SRL

comparisons in *tuple-tuple* similarity matrix is required for an efficient paraphrase classifier. SRL is one of the most basic unit of information with which meaning of a text can be comprehended. Formalization proposed for *tuple* representation and for *tuple* paraphrase recognition are useful for any SRL based NLP task in general.

5 References

[1] Fabio Rinaldi, James Dowdall, Kaarel Kaljurand, Michael Hess, and Diego Molla, "Exploiting paraphrases in a Question Answering system," in *Second international workshop on Paraphrasing, Volume 16*, Sapporo, Japan, 2003, pp. 25-32.

[2] Regina Barzilay and Kathleen R. McKeown, "Sentence Fusion for Multidocument News Summarization," *Computational Linguistics*, pp. 297-328, 2005.

[3] Liang Zhou, Chin-Yew Lin, and Eduard Hovy, "Re-evaluating machine translation results with paraphrase support," in *Conference on Empirical Methods in Natural Language Processing*, Sydney, Australia, 2006, pp. 77-84.

[4] Andrew Finch, Young-Sook Hwang, and Eiichiro Sumita, "Using Machine Translation Evaluation Techniques to Determine Sentence-level Semantic Equivalence," in *Proceedings of 3rd International Workshop on Paraphrasing*, Jeju Island, Korea, 2005.

[5] Yitao Zhang and Jon Patrrick, "Paraphrase Identification by Text Canonicalization," in *Proceedings of Australian Language Technology Workshop*, Sydney, Australia, 2005, pp. 160-166.

[6] Stephen Wan, Mark Dras, Robert Dale, and Cecile Paris, "Using Dependency-Based Features to Take the "Para-farce" out of Paraphrase," in *Proceedings of the Australasian Language Technology Workshop*, Sydney, Australia, 2006, pp. 131-138.

[7] Prodromos Malakasiotis, "Paraphrase Recognition Using Machine Learning to Combine Similarity Measures," in *Proceedings of the ACL-IJCNLP 2009 Student Research Workshop*, Suntec, Singapore, 2009, pp. 27-35.

[8] Martha Palmer, Daniel Gildea, and Paul Kingsbury, "The Proposition Bank: An Annotated Corpus of Semantic Roles," *Computational Linguistics, 31:1*, pp. 71-105, 2005.

[9] Sameer S Pradhan, Wayne Ward, and James H Martin, "Towards Robust Semantic Role Labelling," *Computational Linguistics, Vol 34, No 2*, pp. 289-310, 2008.

[10] Bill Dolan, Chris Quirk, and Chris Brockett, "Unsupervised construction of large paraphrase corpora: exploiting massively parallel news sources," in *20th international conference on Computational Linguistics*, Geneva, Switzerland, 2004, p. Article No. 350.

[11] Long Qiu, Min-Yen Kan, and Tat-Seng Chua, "Paraphrase Recognition via Dissimilarity Significance Classification," in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, Sydney, Australia, 2006, pp. 18-26.

[12] Ion Androutsopoulos and Prodromos Malakasiotis, "A survey of paraphrasing and textual entailment methods," *Journal of Artificial Intelligence Research, 38:1*, 2010.

[13] Dekang Lin, "Automatic retrieval and clustering of similar words," in *17th international conference on Computational linguistics*, Montreal, Quebec, Canada, 1998, pp. 768-774.

[14] Ronan Collobert and Jason Weston, "A unified architecture for natural language processing: deep neural networks with multitask learning," in *International Conference on Machine Learning (ICML)'08*, Helsinki, Finland, 2008, pp. 160-167.

[15] Joseph Turian, Lev Ratinov, and Yoshua Bengio, "Word representations: A simple and general method for semi-supervised learning," in *Association for Computational Linguistics'10*, Uppsala, Sweden, 2010, pp. 384-394.

[16] Richard Socher, Eric H. Huang, Jeffrey Pennington, Andrew Y. Ng, and Christopher D. Manning, "Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection," in *Advances in Neural Information Processing Systems 24*., 2011, pp. 10-18.

[17] Michel Galley and Kathleen McKeown, "Improving word sense disambiguation in lexical chaining," in *International joint conference on Artificial intelligence*, Acapulco, Mexico, 2003, pp. 1486-1488.

[18] George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller, "Introduction to WordNet: An On-line Lexical Database," *International Journal of Lexicography*, pp. 235-244, 1990.

[19] Yoshua Bengio, *Learning Deep Architectures for AI*. Hanover, MA, USA: Now Publishers Inc., 2009.

[20] Ronnan Collobert et al., "Natural Language Processing (Almost) from Scratch," *Journal of Machine Learning Research 12*, pp. 2461-2505, 2011.

[21] Dan Klein and Christopher D. Manning, "Accurate unlexicalized parsing," in *41st Annual Meeting on Association for Computational Linguistics*, Sapporo, Japan, 2003, pp. 423-430.

[22] Michael Collins, "Head-Driven Statistical Models for Natural Language Parsing," *Computational Linguistics*, pp. 589-637, 2003.

SESSION

WEB AND TEXT MINING

Chair(s)

Dr. Peter Geczy
Dr. Robert Stahlbock
Dr. Gary M. Weiss

Analyzing Conflict Narratives to Predict Settlements in EBay Feedback Dispute Resolution

Xiaoxi Xu, David A. Smith, Tom Murray and Beverly Park Woolf
Department of Computer Science, University of Massachusetts, Amherst, MA, USA

Abstract— *We explore the possibility of predicting settlements in online disputes by performing text-analysis on conflict narratives from disputant parties. The experiment domain is eBay Motor vehicles, in which disputants try to resolve complaints, possibly working with online human mediators. The conflict discourse is analyzed based on the divergence of topic distributions in a generative model that extends latent Dirichlet allocation (LDA) to include role information. A set of distance schemes and a heuristic are designed for various negotiation scenarios to predict settlements. We analyze the quality of discovered topics in terms of topic coherence and evaluate settlement classification and prediction power for settlements on unseen data. Experimental results show that this unsupervised model outperforms a state-of-the-art supervised learner on precision, recall, and F-measure. A supervised learner using a few derived features from this model outperforms that using bag-of-word features in terms of precision and efficiency.*

Keywords: text mining; topic modeling; online dispute resolution

1. Introduction

This research focuses on the ability to predict whether two online disputants will reach a settlement based on analysis of their conflict discourse. Automating the process of prediction in online disputes is challenging, in part, because it requires understanding the discourse in negotiation. We developed a latent variable topic model for representing negotiation and prediction that includes a multiple-level hierarchy to represent cases and negotiated exchanges within each case. Moreover, the model represents both topics of disputes and topic usage by each type of disputants. Ultimately, we hope to design an automated dispute resolution process, in which the model can identify interests and positions of disputants and assess their priorities from their negotiations. The present model is based on the assumption that if topics used by disputant parties are aligned, it is likely that a settlement can be reached. Thus we measure the divergence of topic distributions to make predictions about settlements.

This model is tested in the domain of eBay Motors vehicles feedback. Through the gracious generosity of collaborators, including eBay and NetNeutrals [1], we acquired over 4,000 online exchanges among eBay participants involved in sales of automobiles and primarily directed at removing negative feedback, see Table 1. Experiments with this data show that the new dispute model outperforms a state-of-the-art supervised learner on precision, recall, and F-measure. Recall is important for this task because the goal of removing feedback is to remove unwarranted feedback. A mistakenly removed feedback can always be added back on eBay by users, but a delayed unfair feedback will not only mislead other people, it can also ruin a seller's reputation and economic future.

This research makes two contributions: development of a generative model for online dispute discourse and design of a set of distance schemes and a heuristic to analyze conflict narratives and predict agreement. The organization of the paper is as follows. In Section 2, we introduce the concept of online dispute resolution and the experimental domain. In Section 3 we describe the generative model and its Gibbs sampler. Section 4 introduces the experimental setup followed by experimental results in Section 5. We discuss related work in Section 6 and conclude with future plans in Section 7.

2. EBay Feedback Disputes

People doing business at online auction markets (e.g., eBay) are inevitably anxious about their transactions. Buyers and sellers usually engage in one-shot deals meaning that they have no prior relationship before the transaction and do not anticipate any future commercial relationship [1]. "Relationshipless" disputes reduce the trust between two parties which is the root of their anxiety. In order to solve this public anxiety problem, eBay puts in place a reputation system for buyers and sellers to build trust, that is, the feedback mechanism. The use of feedback rating and comments is a way for buyers and sellers to judge the conduct of the other party for any transaction. Feedback is visible to all users and therefore would influence sellers' or buyers' future business. Although acquiring a positive feedback is important, avoiding a negative one requires exercising more care. This is because if sellers ignore the negative feedback, they risk harming future online sales.

2.1 Dispute Process

NetNeutrals is an Online Dispute Resolution (ODR) program that manages disputes or disagreements online. The

[1] http://www.juripax.com and http://www.netneutrals.com/

company has been contracted by eBay to review Motors feedback disputes. Nearly all the disputes are about negative feedback placed on a seller's website by the buyer. Neutrals are trained, independent professionals with automotive service experience. As an online dispute resolution program, NetNeutral offers eBay users two types of voluntary service to resolve customer disagreements. Direct Negotiation is a free dispute resolution process in which two disputant parties work together to come to a resolution on their own without the help of a third party. The independent Feedback Review (IFR) is a dispute resolution process that costs $100 where a third party determines whether a rating should be descored. The IFR evaluates evidence provided by buyers and sellers and offers comments based on eBay's guidelines, including did the member demonstrate a good faith effort to complete the transaction? was the feedback submitted in a reasonable timeframe? is the transaction-related information factually inaccurate? did the member make an attempt to extract excessive value from the other party?

3. A Generative Model for ODR

This research reduces the online dispute into a binary classification problem and presents a language model to predict settlements of disputes based on disputants' narratives. Such an automated process should be advantageous over that of a human reviewer in that it would be more consistent in the manner of judgment, more impartial, efficient, and cost-saving. In future work we hope to enhance the model so that it also recognizes participants' interests and positions, assesses priorities from their negotiations, provides interventions at the proper time, and computes resolutions that may provide each side with more than they themselves might be able to negotiate [1].

To model the negotiation process among disputants and predict case resolutions, we propose a disputant negotiation model (DNM) that extends LDA [2] to include role information. The model predicts dispute resolutions based on evaluating the divergence of disputants' topic distributions. The new DNM model does not have a label node that represents case resolutions, since we are exploring how to represent the divergence of topic distributions from the perspective of a generative process, which is a challenging yet unexplored research problem and label information about case resolutions is not necessarily available in the real world.

3.1 Disputant Negotiation Model (DNM)

The graphical representation of DNM is shown in Figure 1. In DNM, the outermost plate denotes a dispute case or session. Each session contains a number of exchanges among disputants. DNM assumes the following generative process for our dispute corpus:

1. For every topic ϕ out of K, draw a word distribution $\phi_k \backsim$ Dirichlet (β).

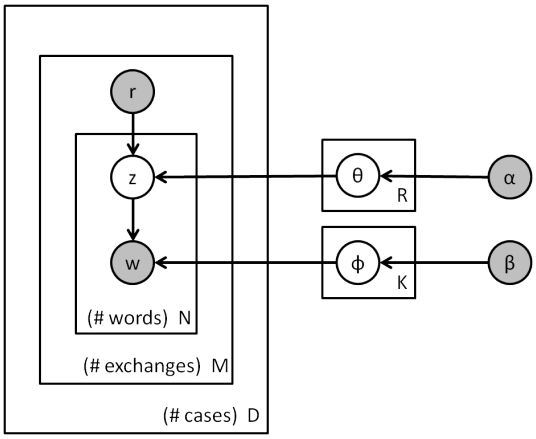


Fig. 1: Disputant Negotiation Model

2. For each disputant r, draw a topic proportion $\theta_r \backsim$ Dirichlet (α).

3. For each exchange m in each case d,

(1) Observe the disputant that generates the exchange.

(2) For each word,

(a) Draw $Z_{d,m,n} \backsim$ Multinomial ($\theta_{r_{d,m}}$).

(b) Draw $W_{d,m,n} \backsim$ Multinomial ($\phi_{z_{d,m,n}}$).

3.2 Gibbs Sampling for DNM

We use collapsed Gibbs sampling [3] to estimate the posterior distribution of hidden variable z given the input variables $\mathbf{w}$, and $\mathbf{r}$, and model parameters, α and β.

$$P(\theta, \phi, z|\mathbf{w}, \mathbf{r}, \alpha, \beta) = \frac{P(\theta, \phi, z, \mathbf{w}, \mathbf{r}|\alpha, \beta)}{P(\mathbf{w}, \mathbf{r}|\alpha, \beta)}$$

Note that we use symmetric Dirichlet priors α, β, in this work, and it is easy to adapt to use asymmetric Dirchlet priors in our model.

Using Gibbs sampling, we construct a Markov chain that converges to the posterior distribution on z and then use the results to infer θ and ϕ. The transition between successive states of the Markov chain is achieved from random sampling z from its distribution conditioned on all other variables, summing out θ and ϕ. By derivation, we get:

$$P(z_i|\mathbf{z}_{-\mathbf{i}}, w, r) \propto \frac{N_{k|r} + \alpha}{N_r + K\alpha} \cdot \frac{N_{w|k} + \beta}{N_k + V\beta}$$

where the subscript z_{-i} denotes all topic assignments excluding the ith word. $N_{k|r}$ is the number of times that topic k is assigned to disputant r, excluding the current instance, and $N_{w|k}$ is the number of times that word w is assigned to topic k, excluding the current instance.

After the Gibbs sampling process, the model parameters in DNM can be obtained as follows:

$$\phi_{w|k} = \frac{N_{w|k} + \beta}{N_k + V\beta} \qquad \theta_{k|r} = \frac{N_{k|r} + \alpha}{N_r + K\alpha}$$

Table 1: Properties of the data set

of 327 Cases; 3982 Exchanges
792 Disputants; Average of 12 posts/case
eBay offers over six million goods and services for sale every day and assumes little or no responsibility for the transactions. When problems arise, members write negative feedback about the other party. eBay handles *40-60 million online disputes* per year. Bad feedback primarily hurts sellers. We examined some of the 20% of the online disputes that require human facilitators.

Table 2: Data statistics with various scenarios

Scenarios	Feedback Removed	Feedback Remained
Mediator, Buyer and Seller	88	50
Buyer and Seller	12	45
Mediator and Buyer	1	1
Mediator and Seller	110	20
Total	211	116

where $\phi_{w|k}$ is the probability of using word w in topic k, and $\theta_{k|r}$ is the probability of using topic k by disputant r.

4. Experimental Setup

In this section, we describe the eBay Motors feedback dispute data set and how we devised distance schemes to measure topic distributions under various scenarios [2].

4.1 Data Set

The eBay Motors data set is a collection of discourses of 2-3 people in conversation around removing negative feedback from 2005 to 2008 [3]. Table 1 summarizes the properties of this data set. Each of the 327 cases falls into one of the following 4 scenarios.

- **Scenario 1: Mediator, Buyer and Seller**
- **Scenario 2: Buyer and Seller (no mediator)**
- **Scenario 3: Mediator and Buyer (Seller did not participate)**
- **Scenario 4: Mediator and Seller (Buyer did not participate)**

The negotiation process associated with each scenario is shown in Figure 2. We further provide data statistics for various scenarios in Table 2. The cases that include Mediator, Buyer and Seller represents 42.20% of all the cases, Buyer and Seller (no mediator) represent 17.43% of the cases, Mediator and Buyer represent 0.61% and Mediator and Seller represent 39.76% of the cases. Furthermore, 64.53% of the cases in this data are successfully settled, while 35.47% of the cases remain unsettled.

4.2 Distance Schemes for Various Scenarios

The idea of using the divergence of topic distributions through text analysis to predict a resolution to a dispute

[2]This research is part of a larger project using text analysis in the domains of deliberative communication [4] [5].

[3]After 2008, NetNeturals changes their procedure to move toward an arbitration model.

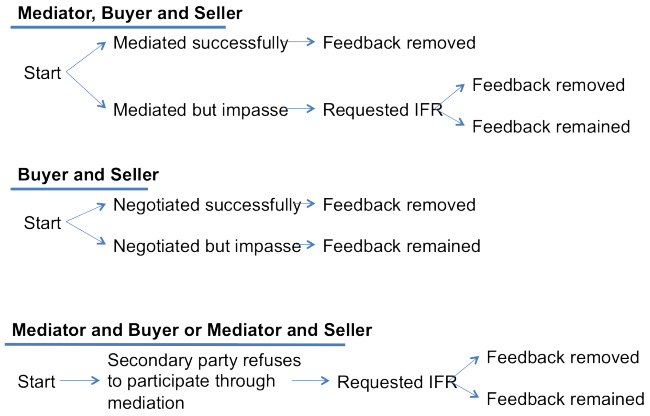


Fig. 2: An illustration of negotiation processes for various scenarios. All participants are human, including mediator, buyer, seller, and independent feedback review (IFR).

is based on the following assumption: *Lower divergence correlates with increased possibility of a resolution (which means feedback removal in the case of eBay disputes).*

Note that an IFR may be requested to evaluate the situation when disputants reach an impasse, and then a mediator will inform disputant parties of the outcome. This means that the content of discourses from mediators has the information of dispute outcomes, which will provide supervisory information for the model. We thus do not use topic distributions from mediators for the settlement prediction task.

We now provide three distance schemes (DS) and one heuristic for the four scenarios provided in the previous section.

DS1 for Mediator, Buyer and Seller

D_1 = MIN (x, y)

where x = Div(Buyer's topic distribution, Seller's topic distribution), y = Div(Mean(Buyer's topic distribution, Seller's topic distribution), guideline's topic distribution), and Div is a divergence metric that will be introduced later.

For scenario 1 (Mediator, Buyer and Seller), the case resolution can be either mediated successfully or mediated but remain at impasse. We develop two distance measures corresponding to these two situations. The distance used for predicting settlement will take the minimum. For the cases that are mediated successfully, only the divergence of the buyer's topic distribution is compared against the seller's topic distribution. For the cases that are mediated but result in an impasse, the average of topic distributions from the two disputant parties is used and compared with the topic distribution of the eBay feedback guidelines.

DS2 for Buyer and Seller

D_2 = Div(Buyer's topic distribution, Seller's topic distribution)

In scenario 2 (Buyer and Seller), negotiations always occur between disputants, regardless of the outcome. There-

fore, we only need to evaluate the divergence between buyer's topic distribution and seller's topic distribution.

DS3 for Mediator and Buyer or Mediator and Seller
D_3 = Div(Buyer's or Seller's topic distribution, guideline's topic distribution)

We also describe a heuristic:

Heuristic for Mediator and Buyer/ Mediator and Seller
# of exchanges (posts) in a case

As explained above, the data from scenarios (Mediator and Buyer) and (Mediator and Seller) have missing information. To deal with the missing data issue, we designed a heuristic in addition to a distance measure that is not reliable when used alone. The distance measure evaluates the divergence between buyer's or seller's topic distribution and the topic distribution of the eBay feedback guidelines. The heuristic was developed based on an assumption about the structure of the negotiation process (i.e., the number of interactions/posts): *More interactions lead to a settlement (or feedback removed in eBay disputes).* Note that the heuristic of using post numbers for prediction is only used for scenarios (Mediator and Buyer) and (Mediator and Seller).

We used two different methods to measure distributional similarity: symmetric Kullback Leibler divergence [6] and Jensen-Shannon divergence [7]. Assume that P and Q are two topic distributions.

The symmetric Kullback Leibler divergence is given by:

$$SKLD(P||Q) = \frac{D_{KL}(P||Q) + D_{KL}(Q||P)}{2}$$

where $D_{KL} = \sum_i P(i) log \frac{P(i)}{Q(i)}$.

The Jensen-Shannon divergence based on Kullback Leibler divergence is given by:

$$JSD(P||Q) = \frac{D_{KL}(P||M) + D_{KL}(Q||M)}{2}$$

where $M = \frac{P+Q}{2}$.

We preprocessed the data by filtering standard English stopwords and tokens of less than two characters. We used unigram features and the Porter stemmer [4]. After data preprocessing, we had 134,184 words with vocabulary size 3194. It is not surprising that we have a small size of vocabulary given that the dispute discourse is from a single domain. We experimented with different configurations of the number of topics and found that three topics provided a good overview of the contents of the corpus. The Dirichlet priors alpha was set to 16, beta to 0.1; the Gibbs sampler was run with 1000 burn-in and 1000 sampling iterations.

5. Results

We performed three sets of experiments to evaluate the proposed model. In the first experiment, we evaluated the topics discovered by DNM, in the second we assessed the performance of DNM on the task of settlement classification and in the third we tested the predictive power of DNM on unseen data. The results below use a single sample from the Gibbs sampler.

[4]http://tartarus.org/martin/PorterStemmer/

Topic 0 *Transaction*	
WORD	**PROB.**
feedback	0.0729
post	0.0426
guidelin	0.0355
rate	0.0337
review	0.0303
withdraw	0.0291
case	0.0260
meet	0.0221
transact	0.0208
ebai	0.0189

Topic 1 *Subject Matter*	
WORD	**PROB.**
car	0.0331
vehicl	0.0210
seller	0.0152
buyer	0.0150
state	0.0103
time	0.0093
purchas	0.0083
said	0.0081
ebai	0.0080
item	0.0075

Topic 2 *Mediation*	
WORD	**PROB.**
feedback	0.0251
thank	0.0194
want	0.0162
mediat	0.0160
go	0.0156
pleas	0.0151
know	0.0144
neg	0.0140
ask	0.0136
work	0.0133

Fig. 3: General topics as discovered by DNM in the eBay dialogues and the top 10 words related to those topics. (Note that the word "eBay" becomes "ebai" after stemming.)

5.1 Topic Discovery and Quality Evaluation

Figure 3 illustrates the three topics learned by the DNM model for the eBay dispute corpus. The topics were extracted from a single sample at the 2000th iteration of the Gibbs sampler. Each topic is illustrated with the top 10 words most likely to be generated conditioned on the topic. The first topic is mostly related to *transaction* (e.g., feedback, post, review); the second topic is related to the *subject matter* (e.g., car, seller, purchase); and the third topic is related to *mediation* (e.g., mediate, thank, want). In a closer examination, we found that 30% of the text was categorized as *transaction*, 43% as *subject matter*, and 27% as *mediation*.

5.1.1 Topic Coherence

Perplexity [8] is often used for evaluating model performance on unseen data. But practically, we are interested in whether learned topics are coherent, that is, whether words in a topic are semantically related to any other words in the same topic. In this work, we used the topic coherence metric [9] to evaluate the quality of learned topics. The assumption of topic coherence is that pairs of words belonging to a single topic will co-occur within a single document, whereas those belonging to different topics will not. In other words, more words will co-occur in coherent topics; few words will co-occur in random topics.

The topic coherence metric is defined as:

$$TC(k; W^{(k)}) = \sum_{m=2}^{M} \sum_{i=1}^{m-1} log \frac{D(w_m^{(k)}, w_i^{(k)}) + 1}{D(w_i^{(k)})}$$

where $D(w)$ is the document frequency of word w and $D(w, w\prime)$ is the co-document frequency of word v and $v\prime$, and $W^{(k)}$=$(w_1^{(k)}, \ldots, w_M^{(k)})$ is a list of the M most probable words in topic k. A smoothing count of 1 is included to avoid taking the logarithm of zero. The coherence scores

Table 3: Coherence of learned topics using the 5 most salient words

Topics	Scores	5 Most Salient Words
Topic 0	-59.4	feedback, post, guidelin, rate, review
Topic 1	-58.0	car, vehicl, seller, buyer, state
Topic 2	-64.2	feedback, thank, want, mediat, go

Table 4: Coherence of learned topics using the 10 most salient words

Topics	Scores	10 Most Salient Words
Topic 0	-212.1	feedback, post, guidelin, rate, review, withdraw, case, meet, transact, parti
Topic 1	-242.2	car, vehicl, seller, buyer, state, time, purchas, said, ebai, item
Topic 2	-240.6	feedback, thank, want, mediat, go, pleas, know, neg, ask, work

of learned topics using the 5 most salient words are shown in Table 3, and those using the 10 most salient words are shown in Table 4. Numbers closer to zero indicate higher coherence. As can be seen from Table 3, the learned topics are highly coherent.

5.2 Settlement Classification

In this section, we present the results of settlement classification by our unsupervised model DNM and also compare its performance with Support Vector Machine (SVM) [10], a state-of-the-art supervised learner for text classification. The classification performance is evaluated quantitatively in terms of Accuracy (% of correct predictions on resolved cases), Precision (% correct of cases that were settled), Recall (% labeled as "settled" that were predicted to be settled), and F-measure (the harmonic mean of precision and recall). As explained earlier, reputation is a precious commodity on eBay. If an automated system such as DNM can achieve high precision and recall then unfair feedback that negatively impacts users can be efficiently removed.

We experimented with two divergence metrics to measure the divergence of topic distributions and found that the following thresholds work best in the Motors domain: (1) if the symmetric Kullback Leibler divergence (SKLD) of the topic distribution is below 0.1, the case is considered settled; (2) if the Jensen-Shannon divergence (JSD) of the topic distributions is below 0.02, the case is considered settled; (3) if the number of exchanges (interactions) in a case is more than 5, the case is considered as settled [5].

Figure 4 shows the classification performance of DNM by using (1) divergence metrics alone (left panel), and (2) divergence metric together with the number of posts (right panel). Please note that the heuristic was only applied to scenarios 3 and 4. The upper left table shows the performance of using SKLD; the upper right table shows that of using SKLD with post numbers. It is expected that the classification performance is boosted by using the heuristic, because it

[5]The heuristic of using post numbers for prediction is only used for scenarios (Mediator and Buyer) and (Mediator and Seller).

Symmetric Kullback Leibler Divergence (SKLD)

True Positive = 88	False Positive = 50
False Negative = 123	True Negative = 66

Accuracy = 47.10%
Precision = 63.77%
Recall = 41.71%
F-measure = 50.43%

Scenarios	Precision	Recall
Mediator, Buyer, Seller	51.13%	69.23%
Buyer and Seller	83.33%	31.25%
Mediator and Buyer	0%	0%
Mediator and Seller	30.00%	80.49%

SKLD + Postnum

True Positive = 166	False Positive = 62
False Negative = 45	True Negative = 54

Accuracy = 67.28%
Precision = 72.81%
Recall = 78.67%
F-measure = 75.63%

Scenarios	Precision	Recall
Mediator, Buyer, Seller	51.13%	69.23%
Buyer and Seller	83.33%	31.25%
Mediator and Buyer	100%	100%
Mediator and Seller	100%	85.27%

Jensen-Shannon Divergence (JSD)

True Positive = 117	False Positive = 69
False Negative = 94	True Negative = 47

Accuracy = 50.15%
Precision = 62.90%
Recall = 55.45%
F-measure = 58.94%

Scenarios	Precision	Recall
Mediator, Buyer, Seller	71.59%	67.74%
Buyer and Seller	83.33%	25%
Mediator and Buyer	0%	0%
Mediator and Seller	40%	83.02%

JSD + Postnum

True Positive = 184	False Positive = 81
False Negative = 27	True Negative = 35

Accuracy = 67.00%
Precision = 69.43%
Recall = 87.20%
F-measure = 77.31%

Scenarios	Precision	Recall
Mediator, Buyer, Seller	71.59%	67.74%
Buyer and Seller	83.33%	25%
Mediator and Buyer	100%	100%
Mediator and Seller	100%	84.02%

Fig. 4: Performance of DNM for settlement classification by using (1) divergence metrics alone (left panel), and (2) divergence metric combined with post number (right panel)

accounts for the effect of applying distance measure on data with missing information. Similarly, the performance of JSD together with the heuristic is better than using JSD alone. When comparing the performance of the use of different divergence metrics (i.e., the upper left table and the lower left table), we found that JSD achieves higher accuracy, recall and F-measure, while SKLD achieves higher precision. Of the four experimental settings, SKLD with post number has greater success for settlement prediction in terms of accuracy and precision, while JSD with post number performs better on recall and F-measure, as highlighted in Figure 4. We also found that, in all of the experimental settings, the proposed model had consistent higher recalls on scenarios that involve a mediator (except for scenario 3 that has only one case) than that did those without a mediator. This is because working with a mediator, disputants tend to have focused discussions on the same topics. Therefore, the DNM model more likely correctly predicts the "settled" cases (i.e., feedback removal in the case of eBay disputes), resulting in high recall.

We also compared DNM with SVM, a state-of-the-art supervised learner for text classification. The idea of SVM is that input vectors are non-linearly mapped to a high-diminutional feature space where a linear decision surface can be constructed [11]. The Motors data set is unbalanced because the size of the positive labeled data is twice as large as that of the negative labeled data. In order to effectively run SVM, we split the data into 2 subsets and preprocessed the data in a similar way as we did for DNM. The performance of SVM that uses unigram features (term occurrence), linear kernel, with 5-fold and 10-fold cross validations are shown in Figure 5. We also tested other kernels, but found that

SVM (5-Fold CV)	
True Positive = 80	False Positive = 42
False Negative = 26	True Negative = 74
Accuracy = 69.30%	
Precision = 65.57%	
Recall = 75.47%	
F-measure = 70.17%	

SVM (10-Fold CV)	
True Positive = 86	False Positive = 42
False Negative = 20	True Negative = 74
Accuracy = 72.11%	
Precision = 67.19%	
Recall = 81.13%	
F-measure = 73.50%	

(a) Applying SVM to balanced subset 1 (Pos: 106, Neg: 116)

SVM (5-Fold CV)	
True Positive = 82	False Positive = 44
False Negative = 23	True Negative = 72
Accuracy = 69.71%	
Precision = 65.08%	
Recall = 78.10%	
F-measure= 71.00%	

SVM (10-Fold CV)	
True Positive = 83	False Positive = 45
False Negative = 22	True Negative = 71
Accuracy = 69.68%	
Precision = 64.84%	
Recall = 79.05%	
F-measure = 71.24%	

(b) Applying SVM to balanced subset 2 (Pos: 105, Neg: 116)

SVM (5-Fold CV)
Accuracy = 69.51%
Precision = 65.33%
Recall = 78.58%
F-measure= 71.35%

SVM (10-Fold CV)
Accuracy = 70.90%
Precision = 66.02%
Recall = 80.09%
F-measure = 72.38%

(c) Average performance of SVM on 2 balanced subsets

Fig. 5: Performance of SVM for settlement classification

SVM (10-Fold CV)	
True Positive = 175	False Positive = 65
False Negative = 36	True Negative = 51
Accuracy = 69.11%	
Precision = 72.92%	
Recall = 82.94%	
F-measure= 77.61%	

DNM+SVM (10-Fold CV)	
True Positive = 159	False Positive = 52
False Negative = 52	True Negative = 64
Accuracy = 67.89%	
Precision = 75.36%	
Recall = 75.36%	
F-measure = 75.36%	

Fig. 6: Performance of SVM (left panel) and DNM + SVM (SVM using derived features from DNM, right panel) for settlement prediction on unseen data

using non-linear kernels did not improve the performance. This is because the number of features is very large in the Motors data, mapping data to a higher dimensional space would not be necessary and not useful for creating a separating decision boundary. The average performance of applying SVM to the two subsets is presented on the bottom row in Figure 5. DNM outperforms SVM in terms of precision and F-measure, when using SKLD with post numbers. It outperforms SVM in terms of precision, recall, and F-measure, when using JSD with post numbers.

5.3 Settlement Prediction on Unseen Data

To evaluate the predictive power of DNM, we also carried out experiments to train a classifier (SVM) using derived features from DNM, which we refer to as DNM+SVM. Specifically, the derived features include the symmetric Kullback Leibler divergence learned from DNM for each case and a binary feature representing whether the number of posts in a case exceeds the confidence threshold we set. As can be seen from Figure 6, DNM+SVM achieves comparable performance to SVM on predicting settlement. We feel that DNM+SVM is quite promising because using a few derived features is much more efficient than using bag-of-word features.

6. Related Work

Previous research has tested the idea that topic divergence distributions can predict whether participants will reach a settlement, as well as the assumption that low divergence in topic distributions will lead to an agreement [12]. For example, in a speed dating classification task, the divergence of topic distributions of dialogues from a dating pair is used to predict men and women's decisions about whether they want to meet again. However, no prior research has attempted to analyze dispute dialogues with topic models and we are the first to develop a topic model for modeling online negotiation and predicting settlements in dispute resolution.

The author-topic model (ATM) [13] is quite similar to the developed DNM model and both models represent the content of disputes. The difference is that DNM has more levels than does ATM to model the nested structure of cases and exchanges within each case. Additionally, DNM models the topic usage of different types of disputants (i.e., buyers and sellers) rather than that of individual disputant and the role of each disputant is observable at the exchange level (and therefore at the case level). Prior research to extend LDA by incorporating a supervision node in the model, such as [14], [15], and [16], are related to this work. DNM does not have a supervision node partly because we are still exploring how to represent the divergence of topic distributions of disputants from the perspective of a generative process, and partly because the label information is not always necessarily observable in the real world.

Research in Online Dispute Resolution (ODR) uses technology to facilitate the resolution of disputes and has been employed to handle disputes from consumer-to-consumer issues and marital separation to workplace grievance and interstate conflicts [6]. ODR shows great advantages over traditional litigation and has the potential to provide greater flexibility, substantial cost-savings, and higher efficiency. In e-commerce, ODR has gained wide popularity by reducing travel time and providing mediators for those who cannot afford them. Moreover, fully automated online services, such as Cybersettle [7], SettlementOnline [8], and ClickNsettle [9], own huge markets for disputes and have had huge commercial success for disputes that are solely over the amount of monetary settlements. Such systems use simple procedures to compare demands with offers and determine settlements if demands and settlements are within a range [1]. For example, Cybersettle alone claims to have handled more than 60,000 transactions during the period between 1998 and 2003, facilitating settlements for more than $350 million [10]. In contrast, other online dispute ventures that are not automated

[6] http://en.wikipedia.org/wiki/Online_dispute_resolution
[7] http://www.cybersettle.com
[8] http://www.settlementonline.com
[9] http://www.clicknsettle.com
[10] http://www.cybersettle.com/about/factsheet.asp

appear to have had more limited success [17]. As Internet usage continues to expand, e-commerce is growing and the number of disputes from e-commerce will also rise. It has become increasingly necessary to design automatic mechanisms for resolving online disputes beyond monetary settlements.

EBay, the largest online auction site, has 83 million users in the U.S. alone in 2009 and millions of sales opening and closing everyday. The eBay reputation system supports sellers and buyers to acquire mutual trust by enabling feedback, ratings and comments, to be left by buyers and sellers for each other. If disagreements about feedback are not settled automatically by disputant parties, then a trained professional may guide participants to reach solutions. Once a fully automated process for reaching settlements has been developed, it will potentially improve on the use of human mediators as it would be wholly impartial, highly efficient, and involve a low cost.

7. Conclusions and Future Work

In this paper, we proposed a generative model to predict whether a settlement would be reached by disputants in the eBay Motor vehicle corpus. The topics discovered by dispute negotiation model (DNM) were related to transaction, subject mattes, and mediation. The coherence score of each topic using the 5 most salient words showed that the learned topics were highly coherent. In a quantitative evaluation of settlement classification, DNM outperformed SVM on precision, recall, and F-measure. In testing the predictive power of the DNM by using derived features from DNM to train a classifier, DNM + SVM achieved comparable performance to SVM with higher efficiency.

These results are encouraging. The next step for predicting whether an agreement will be reached by disputants is to design a pair of supervised models for settlement prediction. The first model would have a resolution label upstream pointing to a node representing the topic divergence of disputant parties. This model would be based on the assumption that disputant parties come to a negotiation with a predetermined approach about whether they are willing to agree to the settlement (in this case to withdraw a negative rating). We are also interested in the reverse problem that has the resolution label downstream. In that case, we assume that disputant parties have a predetermined approach about topics to be discussed and will wait to see if negotiation can help resolve their conflict.

The ultimate research goal is to design an automated dispute resolution process where the model can identify the interests and positions of disputants and assess their priorities from their negotiations. In future work we will explore such a model using derived psychological, lexical, and cohesion-based features from Coh-Metrix [18] and LIWC [19] methods. The hope is that using bag of derived features would yield performance gains over the bag-of-word features used in this study.

8. Acknowledgments

This research was supported in part by grants from the National Science Foundation (0968536). Any opinions, findings, conclusions, or recommendations expressed in the paper are those of the authors and do not necessarily reflect those of the funding agencies.

References

[1] E. Katsh, J. Rifkin, and A. Gaitenby, "E-commerce, e-disputes, and e-dispute resolution: Learning from ebay and other online communities," *Ohio State Journal of Dispute Resolution.*, 2000.

[2] D. M. Blei, "Introduction to probabilistic topic models," *Communications of the ACM*, 2011.

[3] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *Proceedings of the National Academy of Sciences*, vol. 101, no. Suppl. 1, pp. 5228–5235, April 2004.

[4] B. P. Woolf, T. Murray, X. Xu, L. Osterweil, L. Clarke, L. Wing, and E. Katsh, "Computational predictors in online social deliberation," in *Proceedings of the Sixth International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2012.

[5] T. Murray, B. P. Woolf, X. Xu, S. Shipe, S. Howard, and L. Wing, "Supporting social deliberative skills in online classroom dialogues: Preliminary results using automated text analysis," in *Proceedings of the Eleventh International Conference on Intelligent Tutoring Systems (ITS)*, 2012.

[6] S. Kullback and R. Leibler, "On information and sufficiency," in *Annals of Mathematical Statistics 22 (1)*, 1951, pp. 79–86.

[7] J. Lin, "Divergence measures based on the shannon entropy," in *IEEE Transactions on Information Theory 37 (1)*, 1991, pp. 145–151.

[8] H. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno, "Evaluation methods for topic models," in *the 26th International Conference on Machine Learning*, 2009.

[9] D. Mimno, H. Wallach, E. Talley, M. Leenders, and A. McCallum, "Optimizing semantic coherence in topic models," in *EMNLP*, 2011.

[10] S. Theodoridis and K. Koutroumbas, "Pattern recognition," *IEEE Transactions on Neural Networks*, vol. 19, no. 2, p. 376, 2008.

[11] C. Cortes and V. Vapnik, "Support-vector networks," in *Machine Learning 20(3)*, 1995, pp. 273–297.

[12] D. Jurafsky, R. Ranganath, and D. McFarland, "Extracting social meaning: Identifying interactional style in spoken conversation," in *Proceedings of the North American Association of Computational Linguistics (NAACL 2009)*, 2009.

[13] M. Steyvers, P. Smyth, M. Rosen-Zvi, and T. Griffiths, "Probabilistic author-topic models for information discovery," in *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM Press, 2004, pp. 306–315.

[14] D. M. Blei and J. D. McAuliffe, "Supervised topic models," in *NIPS*, 2007.

[15] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning, "Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora," in *EMNLP*, 2009.

[16] S. Lacoste-julien, F. Sha, and M. I. Jordan, "Disclda: Discriminative learning for dimensionality reduction and classification," 2008.

[17] J. Goodman, "The pros and cons of online dispute resolution: An assessment of cyber-mediation websites," *Duke L. and Tech. Rev.*, 2003.

[18] A. C. Graesser, D. S. McNamara, M. M. Louwerse, and Z. Cai, "Coh-metrix: analysis of text on cohesion and language." *Behavior research methods instruments computers a journal of the Psychonomic Society Inc*, vol. 36, no. 2, pp. 193–202, 2004.

[19] J. W. Pennebaker, R. J. Booth, and M. E. Francis, *Linguistic inquiry and word count (LIWC): A computerized text analysis program*. Erlbaum Publishers, 2001.

Bayesian Model Averaging of Named Entity Extraction Algorithms

P. Kidwell, K. Boakye, J. Guensche, R. Glaser, W. Hanley and T. Lemmond
Lawrence Livermore National Laboratory, Livermore, CA, USA

Abstract— *Automatic information extraction (IE) has emerged as a critical tool for commercial, industrial, and governmental applications that are confronted with an explosive growth of digital information. Within the framework of information extraction a hierarchy of objectives exists, many of which are heavily dependent upon the automatic recognition of people, places, and organizations—or, more specifically, named entities—in text documents. In this paper, we present a probabilistic approach to aggregating the results of multiple existing entity extraction technologies. The key achievements presented include: (i) the ability to quantify uncertainty in individual extractors and their parameter estimates, and (ii) increased robustness to the over-fitting commonly observed when individual extractors are trained and evaluated on data from different sources By utilizing Bayesian Model Averaging (BMA), we develop a coherent, data-driven approach for estimating posterior distributions over extracted entities. We demonstrate our approach on several data sets widely used in named entity extraction. The results compare favorably to existing off-the-shelf approaches in traditional settings, as well as in settings where training data are not representative of data encountered under operational conditions.*

Keywords: Entity extraction, bayesian model averaging

1. Introduction

The explosion in the number of electronic documents (e.g., news articles, blogs, and emails) brought about by the advent of the internet and related technologies has made the automatic processing of text increasingly critical. In particular, systems that perform knowledge discovery based on information extracted from text are of growing interest to commercial, industrial, and governmental organizations, as they support analysis, decision making, and the development of strategies and policies. Since named entities (e.g., persons, places, and organizations) and their relationships often constitute a significant portion of the information content within source text, named entity extraction (NEE) has emerged as a key component of these systems.

The purpose of NEE is to automatically identify references to real-world named entities within structured or unstructured text documents, often as part of a more extensive information extraction and analysis effort. Success in this task depends upon accuracy in both the segmentation of text into entity and non-entity regions, as well as the classification of entity regions according to a prescribed (and often hierarchical) collection of entity types. NEE has received considerable attention from the natural language processing (NLP) and, more specifically, information extraction (IE) communities, as evidenced by competitive evaluation tasks such as the Message Understanding Conference (MUC) [1] and the Conference on Computational Natural Language Learning (CoNLL) [2]. Numerous algorithms have been proposed for NEE and have been incorporated into knowledge systems in both research and operational settings.

In an effort to improve upon these systems, some researchers have investigated techniques for combining multiple "base" extraction algorithms into an "aggregate" extraction algorithm. These include methods such as voting [3], [4], stacking [5], or using classifiers for combination [6]. Results from these efforts have demonstrated that further gains can indeed be obtained by leveraging the respective strengths of different extractors.

In this paper, we introduce an aggregation technique based on the principle of Bayesian Model Averaging (BMA). Using the framework discussed in [7], our BMA-based approach estimates a posterior probability distribution over ground-truth hypotheses (i.e. possible segment label assignments) for a "meta-entity"—a region of text defined by the union over individual extractor entity segmentations. This is accomplished as follows: 1) a meta-entity is constructed from the joint output of the constituent base extractors; 2) a "hypothesis space" consisting of possible label assignments to the meta-entity segments is formed; 3) each extraction (i.e. base or aggregate) algorithm produces a distribution over the hypothesis space; and, finally 4) BMA is used to combine the hypothesis probability estimates produced by each of the algorithms based on the respective model posteriors.

This methodology aims to improve on existing extraction techniques in two respects: 1) reducing the variability in performance by accounting for uncertainty associated with individual model estimates, and 2) increasing robustness to the over-fitting frequently associated with training and evaluating on data from different sources. Moreover, unlike many existing aggregation methods, this approach produces a true posterior distribution over possible "hypotheses", thereby enabling the confidence in the extracted data to be quantified.

To present a comprehensive background, Sections 2 and 3 provide a description of the major categories of entity extraction algorithms and common combination techniques, respectively. Section 4 describes the BMA approach and its application to NEE, followed by a discussion of model

estimation and implementation in Section 5. Experimental results are presented and discussed in Section 6, with conclusions and future directions following in Section 7.

2. Entity Extraction Algorithms

Although the substantial investments made by the NLP and IE communities in NEE have generated numerous approaches for solving this problem, these diverse methods can be roughly grouped into a few major categories. These categories include rule-based approaches as well as supervised, semi-supervised, and unsupervised learning methods. In this section, we provide a brief overview of their respective characteristics.

In a rule-based NEE system, entities are identified via a set of rules typically triggered by lexical, syntacical and grammatical cues. These rules are often hand-crafted using linguistic or corpus-based knowledge, and the triggering process is modeled as a finite-state transducer. A simple example of this approach is template matching via regular expressions. While such an approach can be effective and robust to shifting operational conditions, in cases where sufficient representative data exist, rule-based systems are typically outperformed by statistical learning approaches.

Supervised learning—the current state-of-the-art paradigm for NEE—utilizes features derived from text to infer decision rules that attempt to correctly identify and classify entities. Positive and negative examples of entities used to train the algorithm are obtained from a large collection of manually annotated documents. The particular learning algorithm employed varies based upon application-specific limitations and/or specifications, but the most widely accepted include support vector machines (SVMs), decision trees (DTs), hidden Markov models (HMMs), maximum entropy models (MEMs), and conditional random fields (CRFs).

While supervised learning methodologies generally perform quite well in an ideal operating environment (i.e., having plentiful representative data for training), they tend to be highly vulnerable to evolving or sparse data conditions. Semi-supervised (or “weakly supervised") and unsupervised methods attempt to address these issues by circumventing the need for extensive manual annotation.

Specifically, semi-supervised learning is generally an iterative procedure in which a small number of labeled “seed" examples are used to initiate the learning process. The algorithm subsequently generates new training examples by applying the learning from the previous step to unannotated data. The process is repeated until no new examples are generated. One typical approach involves identifying contextual clues from the seed examples and attempting to find new examples that appear in similar contexts. New context information and additional examples are then obtained in an iterative fashion.

Unsupervised learning algorithms, on the other hand, require no annotated data for training. Generally they rely on clustering methods to group named entities based upon similarity of context. Alternative approaches rely on external lexical resources, lexical patterns, and on statistics computed over a large unannotated corpus.

3. Combination techniques

With the variety of extraction algorithms available, a natural extension to traditional NEE approaches is to combine these algorithms—and, consequently, their underlying models—in an attempt to achieve improved performance. The expectation is that these algorithms will collectively use rich and diverse feature representations and will possess complementary characteristics that can be leveraged to enhance positive attributes (e.g., low false alarm or miss rates) while mitigating their individual weaknesses. The most straightforward and intuitive of such approaches utilizes a voting mechanism. Voting techniques examine the outputs of the various models and select the classification with a weight exceeding some threshold. Variations in the voting mechanism employed typically differ in regard to their weighting scheme for individual models. Example voting methods include at-least-N “minority" voting [4] and weighted voting via SVMs [3].

A more sophisticated combination scheme discussed in [?] interpolates a word-conditional class probability distribution across the base extractors $\mathrm{BE}_1^n = \mathrm{BE}_1, \mathrm{BE}_2, \ldots, \mathrm{BE}_n$, where the class, C, corresponds to a word's position relative to a named entity (start/within/end/outside). This distribution, $P(C|w, \mathrm{BE}_1^n)$, is interpolated using weights estimated from training data.

One limitation common to many of these methods is their failure to account for the local context of a word or entity of interest. A CRF model, as proposed by [6], addresses this shortcoming and was shown to yield enhanced performance.

An alternative to the parallel combination techniques described above is the serial process of stacking [5]. In stacking, two or more classifiers are trained in sequence such that each successive classifier incorporates the results of those preceding it. Of course, the above combination approaches can themselves be combined to produce a new methodology, as demonstrated in [8].

Recently, [Lemmond11] proposed a new parallel combination technique based on a “pattern" representation of base extractor output. Specifically, this pattern-based meta-extractor (PME) utilizes a pattern that encodes the joint characteristics of the combined extractor output, D, and (implicitly) their associated errors. The union of overlapping base extractor output regions—the “meta-entity"—provides the textual extent over which a pattern is encoded.

By observing the frequency of these patterns jointly with similar encodings of ground-truth labels for an annotated “evaluation" set, we can compute an estimate of the probability of a hypothesized ground-truth, h, given an observed joint

extractor output d. We then select the hypothesis according to

$$h' = \underset{h \in \Omega}{\operatorname{argmax}}\, p(h|d, \vec{h}, \vec{d}) \quad (1)$$

where Ω is the set of possible hypotheses for a given meta-entity and $p(h|d, \vec{h}, \vec{d})$ is the estimated probability of hypothesis h given an observed output d and the evaluation set $(\vec{h}, \vec{d})$.

One notable property of the PME methodology is that it models the joint characteristics of base extractors and the errors they are likely to produce without knowledge of the underlying algorithms or their individual error processes. As such, each base extractor can be regarded as a "black box" whose output alone is necessary for aggregation. This distinctive characteristic of the PME enables it to address practical issues such as language independence and proprietary restrictions of base extractors. Another notable property is that this method yields a probability estimate for each possible ground-truth hypothesis, facilitating the use of BMA, which is discussed in Section 4.

4. Bayesian Model Averaging

Bayesian Model Averaging (BMA) is a statistical technique designed to account for the uncertainty inherent in the model selection process [9]. This is sharply contrasted with the typical statistical approach in which a single model is selected from a class of models, and fitting proceeds as if this model had generated the data at hand. In NEE, it is common for a single extraction algorithm to be selected a priori and its parameters estimated, or for a collection of algorithms to be combined according to a single aggregation rule. Consequently, NEE represents an appropriate domain for the application of this model averaging technique.

BMA is used to estimate a posterior probability distribution, π, over the value of interest, Δ, given the available data, D, by integrating over a class of models, $\mathcal{M}$, and the model parameters. This can be expressed as

$$\pi(\Delta|D) = \sum_{M \in \mathcal{M}} P(M|D)\pi(\Delta|M, D)$$

where $P(M|D)$ is the model posterior and $\pi(\Delta|M, D)$ is the posterior distribution of the value of interest produced by the model M. Thus, BMA provides a principled mechanism for combining the posterior distributions produced by the individual models by weighting each model in proportion to its posterior probability. Using Bayes' rule, this model posterior can be calculated as

$$P(M|D) = \frac{P(M)P(D|M)}{\sum_{M \in \mathcal{M}} P(M)P(D|M)}. \quad (2)$$

Furthermore, the posterior expectation and variance can be computed as a function of the individual model estimates of the respective quantities. Specifically,

$$E(\Delta|D) = \sum_{M \in \mathcal{M}} P(M|D)E(\Delta|M, D)$$

and

$$\text{var}(\Delta|D) = \sum_{M \in \mathcal{M}} P(M|D)(\text{var}(\Delta|M, D) + E(\Delta|M, D)^2) - E(\Delta|D)^2.$$

5. Models, Estimation, and Implementation

As previously mentioned, the general NEE task consists of both the segmentation of text into entity and non-entity regions and the classification of entity regions according to entity type. Within the meta-entity framework, however, this task reduces to a modified classification problem. More formally, the classification consists of identifying the correct hypothesis h' from the set of possible hypotheses $h \in \Omega$ given the observed output d and the training data $(\vec{h}, \vec{d})$. In our case, we use a maximum a posteriori (MAP) decision rule:

$$h' = \underset{h \in \Omega}{\operatorname{argmax}}\, p(h|d, \vec{h}, \vec{d}).$$

This hypothesis probability estimate is model-dependent. To address the uncertainty inherent in model selection, we can reformulate the estimate within the context of model averaging as

$$p(h|d, \vec{h}, \vec{d}) = \sum_{M \in \mathcal{M}} P(M|\vec{h}, \vec{d})p(h|d, \vec{h}, \vec{d}, M). \quad (3)$$

where the model posterior does not depend upon the newly observed output—i.e., $P(M|\vec{h}, \vec{d}) = P(M|d, \vec{h}, \vec{d})$—and the posterior distribution of h produced by algorithm M is weighted based on the training data.

Aggregating the output of base and/or aggregate extraction algorithms via BMA requires that we specify a model to describe the relationship between extractor output and the underlying truth entity. We begin by assuming that ground truth is generated by the extractor output (by a fixed conditional distribution) where meta-entities are exchangeable within the corpus. This assumption allows a "bag-of-meta-entities" approach similar to the bag-of-words approach of [10] to be employed, with the distinction that, a bag is formed with respect to the corpus rather than an individual document.

First, consider the ground truth h_i and extractor output d_i associated with the i-th of n meta-entities extracted, and denote the evaluation set as $\vec{h} = (h_1, \ldots, h_n)$ and $\vec{d} = (d_1, \ldots, d_n)$. A generative process producing h_i under model M is given by

$$l_i|M \sim \text{Poisson}(\gamma_M)$$

$$d_i|M, l_i \sim \text{Multinomial}(\beta_{Ml_i})$$

$$h_i|M, d_i \sim \text{Multinomial}(\vec{\theta}_{Md_i})$$

where l_i is the length of the i-th meta-entity. That is, a new pair h_i, d_i can be generated by (1) drawing the meta-entity length l_i from a Poisson distribution, (2) drawing the joint extractor output d_i from a multinomial over all joint outputs of a given length, and finally (3) drawing the ground truth h_i from a multinomial conditioned upon d_i. The dimension of the multinomial distribution over ground truth—and, consequently, the parameter vector $\vec{\theta}$—depends upon d_i—specifically, length l_i of the meta-entity determined by d_i. The number of possible representations of the truth under the well-known BIO (begin/inside/outside) model is equal to all sequences of B-I-O where an O can not immediately precede an I. The rate of growth can therefore be described by the recursive formula $a_l = 3a_{l-1} - a_{l-2}$ based upon [11] where l is the length of the meta-entity and $(a_0, a_1) = (1, 2)$.

The overall likelihood for the data $(\vec{h}, \vec{d})$ under model M can be computed according to

$$p(\vec{h}, \vec{d}|\vec{\theta}, \vec{\beta}, \gamma) = \prod_{i=1}^{n} p(h_i|\vec{\theta}_{Md_i}) p(d_i|\beta_{Ml_i}) p(l_i|\gamma_M) \quad (4)$$

where $p(d_i|M, l_i)$ and $p(l_i|M)$ can either be modeled or taken as exogenous in which case they do not contribute to the likelihood. Ultimately, a total of $\sum_{M\in\mathcal{M}} |\mathcal{D}_M|$ multinomial models for $h|d$ must be estimated, where $\mathcal{D}_M$ is the collection of multinomials whose size grows at a rate of a_l^b with b representing the number of constituent extractors whose output is modeled. In practice, meta-entities of length greater than 5 are rarely observed limiting the actual number of fitted models.

Traditionally, there are two primary challenges encountered when implementing model averaging: (1) summing over the possibly large class of models, $\mathcal{M}$; and (2) computing the model likelihood, $P(D|M)$, which involves integrating over all possible model parameter values. In the case of extraction algorithms, however, we only address the latter, as the classes of models considered are small and efficient enough to be readily enumerated and evaluated.

The model likelihood is determined by integrating over all possible parameter values and is given by

$$P(\vec{h}, \vec{d}|M) = \int p(\vec{h}, \vec{d}|M, \vec{\theta}, \vec{\beta}, \gamma) P(\vec{\theta}, \vec{\beta}, \gamma|M) d\vec{\theta} d\vec{\beta} d\gamma. \quad (5)$$

Rather than attempt to evaluate this integral directly, we approximate it by evaluating the likelihood given a point estimate in place of the integral—not an uncommon practice [12]. For example, when h_i is taken as the sole random component then $P(\vec{h}, \vec{d}|M) \approx P(\vec{h}, \vec{d}|M, \hat{\theta})$. One complication of this approach is the potentially varying amount of evaluation data available for estimating the different multinomial models. A simple model likelihood (or log-likelihood) calculation would have the undesirable effect of penalizing models with more evaluation data. Additionally, the exponential dependence of the likelihood on the proportion of correctly classified samples potentially places almost all of the probability mass on a single model [12]. To address this issue, we choose, instead, to compute the mean log-likelihood of the model.

The practical issues of BMA for NEE are not limited to those mentioned above. Additional considerations include parameter estimation, the form of the output of the extraction algorithms, the class of models, and the model priors. These are discussed below.

5.1 Parameter Estimation

Recall from Section 5 that, under the meta-entity framework, a model M consists of a set of multinomial models $\mathcal{D}_M$, each of which has a set of parameters $\vec{\theta}$ that must be estimated. A reasonable approach is to perform maximum likelihood parameter estimation, but difficulties arise when faced with potentially small amounts of training data. To address this, we employ a Bayesian estimate using a non-informative Dirichlet prior $D(\alpha, \ldots, \alpha)$. Using the posterior expectation as the parameter estimate yields

$$\hat{\theta}_{Mdh} = \frac{n_{Mdh} + \alpha}{n_{Md} + a_l\alpha}$$

where n_{Mdh} denotes the number of training examples of under model M, extractor output d, and ground-truth hypothesis h, and $n_{Md} = \sum_{h\in\Omega} n_{Mdh}$. The estimates of $\vec{\beta}$ are similarly obtained.

5.2 Classification

Frequently, the task of classification is separated into two paradigms: 1) Hard classification; and 2) Soft classification. Here we focus on hard classification. Referring to equations 2 and 3, there are two places which require attention in the implementation for BMA: 1) the computation of the model posteriors via model likelihood $P(\vec{h}, \vec{d}|M)$; and 2) the posterior predictive distribution $p(h|d, \vec{h}, \vec{d}, M)$. The latter is easily resolved, as under a hard classification $p(h = h'|d, \vec{h}, \vec{d}, M) = 1$ for the assigned class h' and 0 for all others.

The computation of likelihood $P(\vec{h}|\vec{d}, M, \vec{\theta})$ is almost as easily handled. The likelihood is computed by taking a product of the probability that each training observation is classified correctly. That is,

$$P(\vec{h}|\vec{d}, M, \vec{\theta}) = \prod_{h\in\Omega} \hat{\theta}_{Mdh}^{n_{Mdh}} (1 - \hat{\theta}_{Mdh})^{n_{Md} - n_{Mdh}}$$

where $\theta_{Mdh} = 1 - \epsilon$ and ϵ is an error rate associated with the algorithm [13].

5.3 Model Priors

Two types of priors figure into the BMA framework: 1) $p(\vec{\theta}|M)$, a prior on the parameters given the model; and 2) $p(M)$, a prior distribution over the possible models.

Although non-informative priors are typically desirable for the parameters of a given model, these distributions have been shown to be somewhat less effective when specified over a class of models [14]. As noted in Section 5.1, we use a Dirichlet prior for the multinomial distribution parameters. With regard to the prior distribution over models, we consider several options, discussed below.

1) *Uniform* A uniform prior, $P(M) = 1/|\mathcal{M}|$, over the class of models results in a probability distribution which tends to place more weight on simple models. This is a result of the composition of the model classes and the fact the joint output of more complex models has a higher dimensionality decreasing likelihood.
2) *Complexity based* A prior that places proportionally more weight on the more complicated models can be used to produce a model posterior that more evenly distributes probability over the class of models. In our case, if the joint output space of k extractors grows at a rate of a^k then consider $p \propto a^k$.
3) *Exact Match Rate* An empirical or subjective prior based on the overall performance of a given model can also be used. One reasonable option is

$$P(M) \propto E_M$$

where E_M is the exact match rate, or frequency which the extractor output is identical to the ground truth, associated with model M.

5.4 Model Classes

The class of models $\mathcal{M}$ may be formed in a number of ways, although some of the most interesting focus on addressing a paucity of training data. Typically more complicated aggregation models that account for joint behavior of the constituent extractors require estimating many parameters leading to less reliable estimates than those obtained under simpler frameworks.

1) *Off-the-shelf algorithms* The output of any collection of existing entity-extraction algorithms can be easily handled within the model averaging framework. First, a meta-entity is constructed relative to the joint output of the collection. Second, the output of each algorithm is recorded relative to the joint and the error probabilities are calculated. Finally, a prediction is made on the newly extracted data by evaluating the model posteriors relative to the joint output.
2) *Pattern and likelihood algorithms* The pattern algorithm and likelihood algorithms developed by [7] both use the meta-entity construct and are thus naturally suited to determine the class of models. The performance of these algorithms can vary substantially based upon characteristics of the joint extraction. For example, under the pattern algorithm there are fewer training examples for longer joint outputs, resulting in parameter estimates with higher variability. Within the pattern algorithm framework these problems can be addressed by considering subsets of extractors, or by making independence assumptions., By considering various subsets of extractors a model posterior probability can be calculated which reflects the relative confidence in a specific subset, and similarly for the independence-based approach.
3) *Unions* In general, any combination of model classes can be combined for BMA, provided that the constituent outputs are represented under the meta-entity framework, thereby transforming the problem into one of classification.

6. Experiments

We investigate the performance of BMA from several directions. First, we examine the impact of the tuning parameters and model classes. Second, the performance is compared with other state-of-the-art extraction technologies in a number of different operational settings.

Each experiment was performed utilizing annotated data sets. These datasets include those widely used by the NEE community MUC6, MUC7, and CONLL which are comprised of 7617, 11969, and 26872 entities respectively. It should be noted that although these data sets were manually annotated using a common set of guidelines inter-annotator disagreement produced F-measures of 0.96 to 0.97.

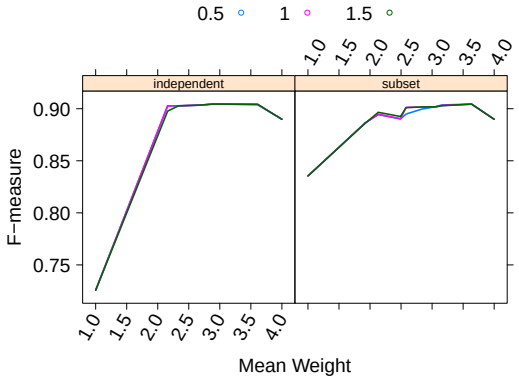
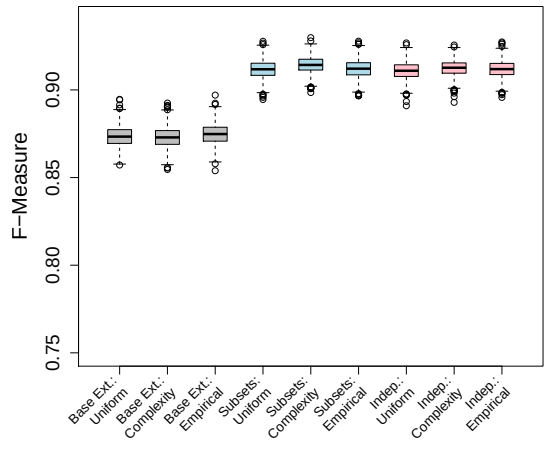
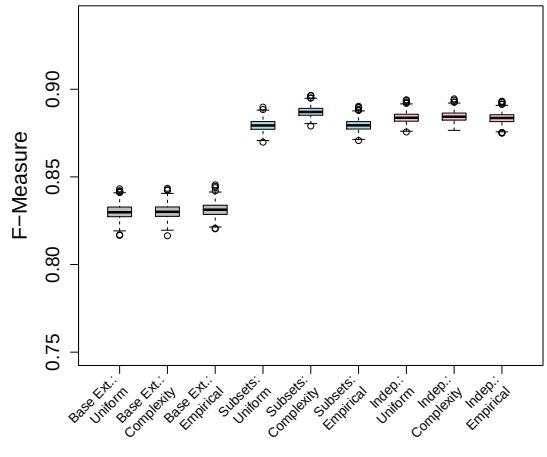


Fig. 1: A comparison of complexity based prior weighting across independence (left) and subset (right) based model classes. The best performance was attained by balancing the prior weights over the range of models. The independence based model class performance was less dependent upon the choice of prior weights, while the the subset class produced the highest f-measure over all priors.

Four off-the-shelf extractors were used in these experiments. They include (1) GATE, a rule-based extraction tool; (2) LingPipe, an extraction tool based on Hidden Markov Models (HMMs); (3) Stanford Named Entity Recognizer (SNER), based on CRFs; and (4) BALIE, an extraction tool that utilizes unsupervised learning. The model fitting and evaluation was based on 10-fold cross validation applied



(a) Train MUC6, Test MUC6



(b) Train MUC6, Test CoNLL

Fig. 2: A comparison of performance for BMA using different model classes and priors.

to the constituent base extraction algorithms. One of two paradigms was used depending upon the nature of the underlying algorithm: 1) Base extraction algorithms whose output was used directly were fit using the standard paradigm; and 2) Aggregation-based algorithms employed a training-training-test split of 45-45-10.

6.1 Model prior and class comparisons

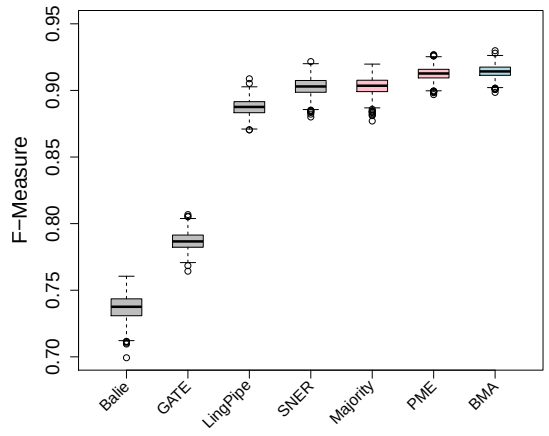
As mentioned in Section 5, the choice of prior distribution and the set of model classes are two practical issues of the BMA approach. Here we explore the implications of these choices with regard to performance in the NEE task. Three model priors and three model classes were analyzed. The model priors were 1) Uniform; 2) Complexity-based; and 3) Empirical based on exact match rate. The model classes included 1) Off-the-shelf (or "base" extractors); 2) Subsets of base extractors; and 3) Different base extractor independence assumptions.

A first step in understanding the behavior of this approach is to study the sensitivity of model averaging to the prior distributions over model class and model parameters. We begin by examining these relationships and inferring the optimal choices for future predictions by training our model on MUC6 and evaluating its performance on MUC7. Figure 1 plots F-measure on the y-axis versus the mean complexity of a given combination of extractors on the x-axis where the mean complexity is computed as $\sum_{\mathcal{M}} P(M)c_M/|\mathcal{M}|$ with c_M denoting the number of base extractors jointly modeled. Instead of exploring all possible forms of $P(M)$ only unimodal functions were considered on the premise that in general either the more complicated or simpler models need to be weighted more heavily. Each line in the two panels represents a combination of model prior α and a model class, independence or subset.

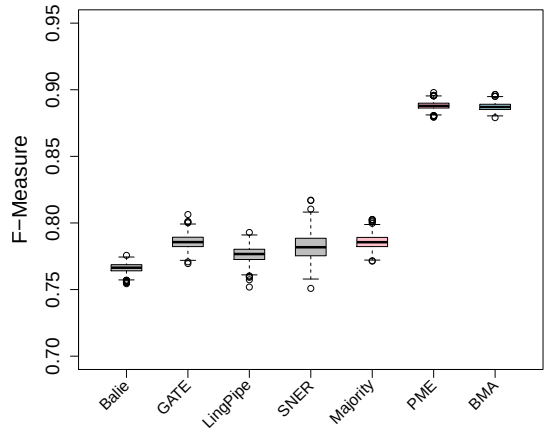
The plots in figure 1 show that the optimal weighting scheme emphasizes the importance of more complicated joint models. When the individual model posterior expectations were relatively even, more complicated models dominated, deferring to the less complicated extractors when insufficient training data were available or the complicated models were poor predictors. Interestingly, the choice of α for the non-informative Dirichlet prior had a very limited effect on f-measure while a highest f-measure was attained at $\alpha = 1.5$, decreasing the alpha to .1 or .01 did correspond to a pronounced decrease in performance. This result may stem from the importance placed on accurate estimation of the most likely hypothesis by the MAP decision rule, i.e. accurate estimation of the probability of the most likely hypothesis is aided by larger values of α although this potentially causes the probabilities of the less likely hypotheses to be overestimated which is of no consequence under this paradigm.

Figure 2 presents boxplots of performance of the various model class and prior combinations as determined by F-score. The boxplots were generated using 1000 bootstrap samples of the weighted F-score from the 10 cross-validation folds. Weighting was determined by the number of ground-truth entities within each fold. Figure 2(a) shows results in which the training and test data originated from the MUC6 data set (the "matched" data condition) and Figure (b) shows results in which training data originated from MUC6 and test data from CoNLL (the "mismatched" data condition).

The first noteworthy result is that a substantial performance difference exists between the base extractor model class and those of extractor subsets and extractor independence. Between the subsets and independence, no notable difference exists. The base extractor models fail to leverage joint information from any of the extractors, and poorer performance results. Also of note, is the performance difference between the matched and mismatched conditions. A moderate degradation is observed when testing on a data set differing from the training data set, though the pairing of the subsets model class and the complexity-based prior seems particularly robust. Lastly, we see that, regarding the model



(a) Train MUC6, Test MUC6



(b) Train MUC6, Test CoNLL

Fig. 3: A comparison of BMA performance to majority rule and the pattern meta-entity extractor (PME).

prior, the complexity-based prior appears to perform as well or better than the other two across both model classes and evaluation conditions.

6.2 Alternate algorithm comparisons

In addition to analyzing differences among the various extraction systems within the BMA framework, analyzing BMA relative to alternate extraction algorithms is naturally of interest. Here we compare a single BMA system—the subsets model class with the complexity-based prior–to a majority rule approach and the basic pattern meta-entity extractor (PME) system. Figure 3 presents these results in the same fashion as above.

For the matched data condition, the performance of majority rule, PME, and BMA are rather similar. Majority rule does appear to lag behind PME and BMA performance is slightly higher than PME, but the difference across systems is small. This is in contrast to the mismatched condition, in which majority rule performance degrades dramatically, but that of PME and BMA do so only moderately.

7. Conclusions

Utilizing bayesian model averaging, we have demostrated an approach to entity extraction which is capable of: (i) reducing the variability in performance by accounting for uncertainty associated with individual models, and (ii) increasing robustness to over-fitting associated with training on a single corpus. In practice, developing priors based on the complexity of the constituent models produced the best results in terms of F-measure. We observed that while model class and selection of a prior are separate components of the process they should be considered simultaneously. Additionally, this approach can be applied to a wide variety extractors and aggregation algorithms as they are all treated as "black boxes".

8. Acknowledgments

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

References

[1] R. Grishman and B. Sundheim, "Message understanding conference-6: A brief history," in *Proceedings of the 16th conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, 1996, pp. 466–471.

[2] E. Tjong Kim Sang and F. De Meulder, "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition," in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, 2003, pp. 142–147.

[3] D. Duong, J. Venuto, B. Goertzel, R. Richardson, S. Bohner, and E. Fox, "Support vector machines to weight voters in a voting system of entity extractors," in *International Joint Conference on Neural Networks*, 2006, pp. 1226–1230.

[4] N. Kambhatla, "Minority vote: at-least-N voting improves recall for extracting relations," in *Proceedings of the COLING/ACL on Main conference poster sessions*. Association for Computational Linguistics, 2006, pp. 460–466.

[5] R. Florian, "Named entity recognition as a house of cards: Classifier stacking," in *proceedings of the 6th conference on Natural language learning-Volume 20*. Association for Computational Linguistics, 2002, pp. 1–4.

[6] L. Si, T. Kanungo, and X. Huang, "Boosting performance of bio-entity recognition by combining results from multiple systems," in *Proceedings of the 5th international workshop on Bioinformatics*. ACM, 2005, pp. 76–83.

[7] T. Lemmond, N. Perry, J. Guensche, J. Nitao, R. Glaser, P. Kidwell, and W. Hanley, "Enhanced named entity extraction via error-driven aggregation," in *Proceedings of 6th ICDM*, 2010.

[8] D. Wu, G. Ngai, and M. Carpuat, "A stacked, voted, stacked model for named entity recognition," in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, 2003, pp. 200–203.

[9] G. Claeskens and N. Hjort, *Model Selection and Model Averaging*. Cambridge University Press, 2008.

[10] Z. Harris, "Distributional structure," *Word*, 1954.

[11] J. Perry, *Number of 31-avoiding words of length n on alphabet 1,2,3 which do not end in 3*. ATT Research Laboratory, 2003.

[12] P. Domingos, "Bayesian Averaging of Classifiers and the Overfitting Problem," in *Proceedings of the 17th ICML*, 2000, pp. 223–230.

[13] M. Kearns and U. Vazirani, *An introduction to computational learning theory*. MIT Press, 1994.

[14] J. Hoeting, D. Madigan, A. Raftery, and C. Volinsky, "Bayesian model averaging: A tutorial (with discussion)," *Statistical Science*, vol. 14, no. 4, pp. 382–417, 1999.

Location-Based Burst Detection Algorithm in Spatiotemporal Document Stream

Keiichi Tamura and Hajime Kitakami
Graduate School of Information Sciences, Hiroshima City University, Hiroshima, Japan

Abstract— *The recent increasing interest in consumer generated media has resulted in numerous studies on extracting topics from documents in micro blogs. These documents are usually arranged in a temporal order and hence are represented as a document stream. This study focuses on a document stream that consists of documents containing creation time and location information. This type of document stream is referred to as a spatiotemporal document stream. We propose a novel algorithm for detecting location-based bursts in a spatiotemporal document stream. To evaluate the new location-based burst detection algorithm, we use an actual spatiotemporal document stream composed of crawling tweets on Twitter. Experimental results show that the algorithm can detect location-based bursts that vary with user location.*

Keywords: spatiotemporal data; text mining; burst detection; consumer generated content; topic detection and tracking;

1. Introduction

With the recent increasing interest in consumer generated media, a large number of documents are continuously created on the Internet. The number continues to exponentially increase specially owing to the widespread use of micro blogs (e.g., Twitter, Facebook, and Google+) for creating online documents [1]. The documents on the Internet are usually arranged a temporal order and hence are represented as a document stream. Topic extraction from a document stream has recently been gaining increasing attention and numerous studies on the text mining domain have been conducted [2], because the contents of these documents include variety types of hot topics such as news, social events, geographical topics, hobbies, and daily happenings.

Kleinberg's burst detection algorithm [3], [4] is one of the most effective techniques to extract topics from a document stream. Kleinberg defines a bursty word as a word that increasingly occurs in a document stream. Some words are highly bursty in the sense that the frequency of their occurrences spike when a particular event attracts public attention. Kleinberg's burst detection algorithm aims to find certain time periods in which there is a high frequency of the occurrence of words. When a word related to an attention-attracting event becomes highly bursty, the interarrival time between documents that include the word becomes smaller during particular time period. Therefore, this time period when a word becomes highly burst can be detected using the interarrival time between the documents.

Recently, the widespread use of smart devices with a global positioning system and geographical applications have resulted in an increase in the number of documents with location information (e.g., geotag). Consequently, many documents in a document stream not only have a creation time but also contain location information. In other words, documents in a docment stream have a spatiotemporal order. The contents of these documents include topics that are closely related to a particular location. Therefore, we need to detect burstiness while considering location information. However, there have been no attention on location-based burst detection algorithms.

If topic "A" is a hot topic in a particular region "B," then it contains useful information in the vicinity of region "B." However, topic "A" is not useful for users far away from region "B." In this case, we need to detect burstiness by considering location. While topic "A" has to be presented as a highly bursty topic for users in the vicinity of region "B," it has to be presented as not highly bursty for users far away from region "B." To satisfy this requirement, it is necessary to integrate location information into burst detection algorithms.

This study focuses on a document stream that consists of documents containing creation time and location information. We call this type of a document stream spatiotemporal document stream (SDS). In this paper, we propose a novel method for detecting location-based bursts in SDS. The main contributions of our study are as follows:

- To enable easy handling of SDS, we define the data model of a document in SDS.
- To detect location-based bursts in SDS, we extend Kleinberg's burst detection algorithm. In our extension, the influence factor of a document is defined as the distance between a user and the location where the document was created. The location-based burst detection algorithm adjusts the burst using the influence factor of the document.
- To evaluate the new location-based burst detection algorithm, we use an actual SDS composed of crawling tweets on Twitter. The experimental results show that the algorithm can detect location-based bursts that vary with user location.

The rest of the paper is organized as follows: Section 2 discusses the related work. Section 3 presents a brief explanation on Kleinberg's burst detection algorithm. Section 4 presents the problem definition of location-based burst detection and a novel method for burst detection in

SDS. Section 5 shows the experimental results. Finally, section 6 concludes this paper.

2. Related Work

Since the Internet gained widespread use, topic detection and tracking [5] has been the most important research area in the text mining domain. In particular, because of the wide spread creation of various online documents on the Internet, there have been many studies on topic detection and tracking in document streams. To detect topics in a document stream that have attracted many people, burstiness is the simplest but the most effective criterion. Therefore, with the increased interest in extracting topics from online documents, such as news, message boards, blogs, micro blogs, several algorithms have been developed to detect bursts in document streams [3], [4], [6], [7], [8], [9], [10], [11].

There are a number of studies on burst detection algorithms. The most significant impact on many studies is Kleinberg's burst detection algorithm [3], [4]. It is based on a queuing theory for bursty network traffic. The details of the algorithm are explained later. It is used for various document streams such as e-mail [3], blogs [11], online publications [12], bulletin board, and social tags [13]. Moreover, there are some studies about the extension of Kleinberg's burst detection algorithm. In particular, Qi He et al. [14] proposed a clustering algorithm for documents in a document stream that uses bursty feature representation as a feature vector for clustering. Leskovec et al. [15] formulated memes as patterns of words by using a scalable clustering approach.

Recently, geographical topic detection and tracking [16], [17], [18], [19] has been attracting increasing attention, because the number of geographical documents have been increasing on the Internet. Sakaki et al. [16] proposed a model for real-time event detection using tweets on Twitter. To detect the location where an event has occurred, they used Kalman filtering and particle filtering, which are widely used for location estimation in ubiquitous computing. Cheng et al. [17] developed a classification method that uses words in tweets with a strong local geo-scope and a lattice-based neighborhood-smoothing model for refining the estimation of a user's location. Yin et al. [18] proposed a method to discover different topics in geographical regions. Furthermore, Yang et al. [19] developed a method to reveal the appearance and disappearance of topics in different regions.

There are numerous studies on burst detection and geographical topic detection and tracking. However, to the best of our knowledge, until now, there is no study that attempts to detect location-based bursts in SDS. This paper describes a data model for SDS and proposes a method for detecting location-based bursts. If location-based bursts can be detected in SDS, we can provide topics that are accurate and helpful for users who want to know local information.

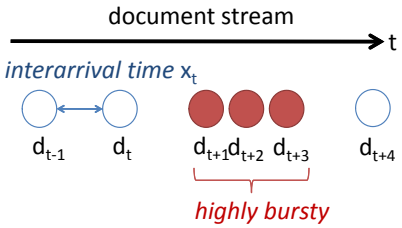


Fig. 1: Example of a Document Stream.

3. Preliminaries

This section presents the definition of a document stream and a burst, and briefly explains Kleinberg's burst detection algorithm.

3.1 Document Stream

A document stream is similar to a data stream. It is defined as a sequence of documents arranged in a temporal order. Fig. 1 shows an example of a document stream. In Fig. 1, the documents are posted in temporal order. The time interval x_t between document d_{t+1} and document d_t is called the interarrival time. Examples of a document stream include, but are not limited to, tweets on Twitter. Tweet i is represented as document d_i. The interarrival time x_i is defined as the time interval between the posting time of tweet $i+1$ and that of tweet i.

3.2 Burst

As the number of documents that include a word related to a particular event increases in a document stream, the interarrival time between these documents becomes smaller. A word is considered highly bursty during a period in which the interarrival time is shorter than usual. In addition, the period is described as bursty. For example, in Fig. 1, the interarrival time between d_{t+1} and d_{t+2}, and that between d_{t+2} and d_{t+3} are smaller than the other interarrival times. In this case, we can observe that this period is highly bursty.

3.3 Kleinberg's Burst Detection Algorithm

Kleinberg defined a model with an infinite-state automaton in which bursts are represented as state transitions. Assuming that there are m states in the automaton, each interarrival time is a probabilistic output that depends on the internal states of the infinite-state automaton. In the model, a state is associated with a burstiness state and a higher state indicates higher burstiness.

Let the sequence of interarrival times between document postings be $x = (x_1, x_1, \cdots, x_n)$. The problem is defined to find an optimal state-transition sequence $s = (s_1, s_2, \cdots, s_n)$ to minimize the following cost function:

$$C(s|x) = \sum_{i=1}^{n-1} \tau(s_i, s_{i+1}) + \sum_{i=1}^{n} (-\ln f_{s_i}(x_i)). \quad (1)$$

The function $\tau(i, j)$ returns a state-transition cost from state i to state j. It is defined as

$$\tau(i,j) = \begin{cases} (j-i)\gamma \ln n, & \text{if } j > i, \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

where $\gamma(> 0)$ is a user-given parameter and n is the number of documents in the document stream being observed. Equation 2 indicates that moving to a higher state incurs a cost and moving to a lower state incurs no cost.

The function $f_k(x_i)$ is the exponential density function for the probability of outputting the interarrival time x_i in state k and defined as

$$f_k(x_i) = \lambda_k e^{-\lambda_k x_i}, \tag{3}$$

where λ_k is the arrival rate of documents associated with state k and is defined as

$$\lambda_k = \frac{n}{T}\beta^k, \tag{4}$$

where n is the number of documents, T is the entire time range and $\beta(> 1.0)$ is a user-given parameter. Equation 4 indicates that a higher state has a higher arrival rate.

The Viterbi algorithm for hidden Markov models, which is a dynamic programming approach, is the most effective solution for determining an optimal state-transition sequence $s = (s_1, s_2, \cdots, s_n)$ to minimize Equation 1. First, we calculate the following cost $C_j(i)$:

$$C_j(i) = -\ln f_j(x_i) + min_l(C_l(i-1) + \tau(l, j)), \tag{5}$$

where $C_j(i)$ is the minimum cost of a state-transition sequence that ends with state j at the i-th time-interval in the document stream. Equation 5 can be calculated using the previous $(i-1)$-th $C_l(i-1)$ $(0 \le l \le m-1)$. Second, we find the minimum cost in $C_j(n)(0 \le j \le m-1)$. Suppose that the minimum cost in $C_j(n)(0 \le j \le m-1)$ is $C_{min}(n)$. Finally, we trace back with $C_{min}(n)$ as the starting point.

4. Location-based Burst Detection

This section presents the problem definition and a novel burst detection algorithm for spatiotemporal document stream (SDS).

4.1 Model and Problem Definition

Suppose that there are n documents in SDS. Let d_i denote the i-th document in SDS; then d_i consists of four items;

$$d_i = <id_i, content_i, ctime_i, cposition_i>, \tag{6}$$

where id_i is the identifier of the document, $content_i$ is the content (e.g., title, textdata, and tags), $ctime_i$ is the creation time of the document, and $cposition_i$ is the location where d_i was created or is located (e.g., latitude and longitude).

Fig. 2 shows an example of SDS comprising six documents. Each document d_i has its own creation time in the time line and a location on the geographical coordinate space.

Let W be a set of all words appearing in SDS. The word time-series data w_i is defined as $w_i = < word_i, CTT_i, CTP_i >$, where $word_i \in W$ is string data, CTT_i is the creation time sequence of the documents that include $word_i$ in their content, and CTP_i is the location sequence of the documents that include $word_i$.

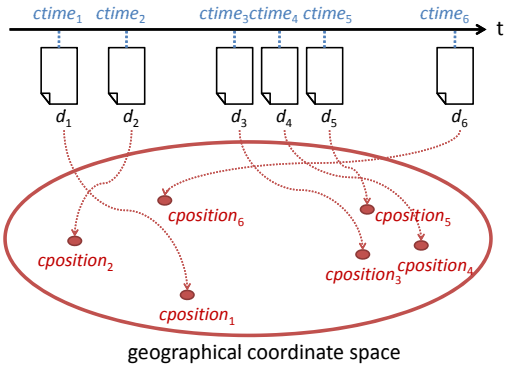


Fig. 2: Example of a Spatiotemporal Document Stream.

$$CTT_i = (ctt_{i,1}, ctt_{i,2}, \cdots, ctt_{i,tnum(i)}), \tag{7}$$

$$CTP_i = (ctp_{i,1}, ctp_{i,2}, \cdots, ctp_{i,tnum(i)}), \tag{8}$$

where $tnum(i)$ returns the number of documents that include $word_i$. The j-th element of CTT_i is represented as $CTT_i[j](= ctt_{i,j})$.

For example, in Fig. 2, suppose that $word_k$ is included in three documents $\{d_3, d_4, d_5\}$. In this case, the creation time sequence of $word_k$ is $CTT_k = (ctt_{k,1}, ctt_{k,2}, ctt_{k,3})$, where $ctt_{k,1} = ctime_3$, $ctt_{k,2} = ctime_4$, and $ctt_{k,3} = ctime_5$. Similarly, the location sequence of $word_k$ is $CTP_k = (ctp_{k,1}, ctp_{k,2}, ctp_{k,3})$, where $ctp_{k,1} = cposition_3$, $ctp_{k,} = cposition_4$, and $ctp_{k,3} = cposition_5$.

Here, let the interarrival time sequence of $word_i$ be $IAT_{CTT_i} = (iat_{i,1}, iat_{i,2}, \cdots, iat_{i,tnum(i)})$, where each $iat_{i,j}$ indicates an interarrival time:

$$iat_{i,j} = \begin{cases} ctt_{i,j} - stime, & \text{if } j = 1, \\ ctt_{i,j} - ctt_{i,j-1}, & \text{otherwise,} \end{cases} \tag{9}$$

$stime$ is the start time of SDS.

The goal of this study is to detect the location-based burst that varies with the user position up. In other words, for each $w_i \in W$, find an optimal state-transition sequence $s = (s_1, s_2, \cdots, s_n)$ to minimize the $C(s|IAT_{CTT_i})$ associated with up.

For instance, suppose that d_3, d_4, and d_5 include the k-th word $word_k$ associated with an topic, and $word_k$ is highly bursty from $ctime_3$ to $ctime_4$ as defined by Kleinberg's burst detection algorithm. Then we need to show users located at a distance from the document creation location that $word_k$ is not highly bursty. This is because distant users are not interested in the topic. In contrast, $word_k$ is highly bursty for users in the vicinity of the document creation location because nearby users would be interested in the topic.

4.2 Main Concept

The simplest intuitive way to find location-based bursts in SDS is to detect bursts from documents that exist around users. Fig. 3 shows an example. There are two users;

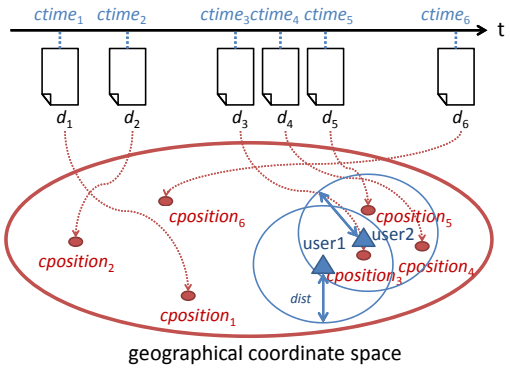


Fig. 3: Cutoff-Distance-Based Burst Detection.

$user1$ and $user2$. Each user uses only the documents that satisfy with $distance(d_i, user) \leq dist$, where the function $distance$ returns the distance between d_i and the $user$. The value of $dist$ is a cutoff distance given as a user-specific parameter. In Fig. 3, there is one document within distance $dist$ from $user1$. Moreover, there are three documents within distance $dist$ from $user2$.

This simple approach using cutoff distance is called cutoff-distance-based burst detection. In this approach, for each $w_i \in W$, we find an optimal state-transition sequence $s = (s_1, s_2, \cdots, s_n)$ to minimize $C(s|IAT_{CTT_i'})$, where CTT_i' is the creation time sequence of documents that satisfy $calc\_distance(ctp_{i,j}, up) \leq dist$. Function $calc\_distance$ is returns the distance between the location of document $ctp_{i,j}$ and the user position up.

Algorithm 1 shows the cutoff-distance-based burst detection algorithm. First, we determine CTT_i' from CTT_i by filtering using the cutoff distance $dist$. Function $append\_sequence(S, item)$ appends $item$ to the tail of sequence S. Second, we generate the interarrival time sequence $IAT_{CTT_i'}$. Finally, we find an optimal state-transition sequence $s = (s_1, s_2, \cdots, s_n)$ to minimize $C(s|IAT_{CTT_i'})$ using function KBD.

Although the cutoff-distance-based burst detection is the easiest way to detect bursts around users, it is largely dependent on the cutoff distance. For example, suppose that word k is highly bursty from $ctime_3$ to $ctime_5$ as shown in Fig. 3, and $user1$ and the $user2$ are close. However, the cutoff-distance-based burst detection shows $user1$ that word k is not highly bursty because there is only one document within $dist$ that include word k. This issue can be avoided by setting a large value for the cutoff distance $dist$. This results in another issue: burst detections are visibly affected by documents far away from users.

To address this issue, we integrate the influence factor of a document into Kleinberg's burst detection algorithm. The influence factor of a document is defined as the distance between a user and the location. The interarrival times are corrected using by the influence factors of documents. Interarrival time is the main factor for state transitions in Kleinberg's burst detection algorithm. Therefore, we correct the sequence of inter-arrival time $x = (x_1, x_2, \cdots, x_n)$ in accordance with the influence factors of documents.

Algorithm 1: Cutoff-Distance-Based Burst Detection

input : cutoff distance $dist$, position of the user up, word time-series data w_i, and parameter list for burst detection $params$
output: optimal state-transition sequence S

$CTT_i' \leftarrow ()$ /* make a empty sequence */
for $j \leftarrow 1$ **to** $|w_i{->}\ CTT_i|$ **do**
  $ctp \leftarrow w_i{->}\ CTP_i[j]$
  if `calc_distance` $(ctp, up) \leq dist$ **then**
    $CTT_i' \leftarrow$ `append_sequence` (CTT_i', ctp)
$IAT_{CTT_i'} \leftarrow ()$ /* make a empty sequence */
for $j \leftarrow 1$ **to** $|CTT_i'|$ **do**
  if $j = 1$ **then**
    $pctt \leftarrow stime$
  else
    $pctt \leftarrow CTT_i'[j-1]$
  $iat \leftarrow CTT_i'[j] - pctt$
  $IAT_{CTT_i'} \leftarrow$ `append_sequence` $(IAT'_{CTT_i'}, iat)$
$s \leftarrow$ `KBD` $(IAT_{CTT_i'}, params)$
return s

4.3 Algorithm

The location-based burst detection algorithm, unlike the cutoff-distance-based approach, does not filter documents according to distance. It corrects the sequence of interarrival time IAT_{CTT_i} by using the influence factors of documents including $word_i$. To correct the interarrival time sequence $x = (x_1, x_2, \cdots, x_n)$, time is added to each interarrival time x_i in accordance with the distance between document d_i and the user. As a result, the interarrival times of documents created far away from the user become longer than their actual interarrival times.

We define the corrected interarrival time as follows:

$$iat'_{i,j} = \begin{cases} ctt_{i,j} - stime + \delta(ctp_{i,j}, up), & \text{if } j = 1, \\ ctt_{i,j} - ctt_{i,j-1} + \delta(ctp_{i,j}, up), & \text{otherwise,} \end{cases} \quad (10)$$

where function δ returns a correction value.

Algorithm 2 shows the algorithm for location-based burst detection. The algorithm uses all the documents that include $word_i$. First, we generate the interarrival time sequence IAT'_{CTT_i} using by Equation 10. Second, we find an optimal state-transition sequence $s = (s_1, s_2, \cdots, s_n)$ to minimize $C(s|IAT'_{CTT_i})$ using function KBD.

There are two methods for interarrival time correction. One is time-difference-based correction and the other is forgetting-factor-based correction. These two correction methods are described as follows:

Time-Difference-based Correction

In this correction, time difference is used for the calculation of correction value. The function $calc\_distance$

Algorithm 2: Location-Based Burst Detection

input : cutoff distance $dist$, user position up, word time-series data w_i, and parameter list for burst detection $params$
output: optimal state-transition sequence S

$IAT'_{CTT_i} \leftarrow ()$ /* make a empty sequence */
for $j \leftarrow 1$ **to** $|w{->}\, CTT_i|$ **do**
  if $j = 1$ **then**
    $pctt \leftarrow stime$
  else
    $pctt \leftarrow w{->}\, CTT_i[j-1]$
  $iat \leftarrow w{->}\, CTT_i[j] - pctt + \delta(w_i{->}\, CTP_i[j], up)$
  $IAT'_{CTT_i} \leftarrow$ `append_sequence` (IAT'_{CTT_i}, iat)
$s \leftarrow$ `KBD` $(IAT'_{CTT_i}, params)$
return s

returns the distance between location dp of the document and user up and SP is Earth's rotation rate.

$$\delta(dp, up) = \frac{calc\_distance(dp, up)}{SP} \tag{11}$$

Forgetting-Factor-based Correction

In this correction, a forgetting factor[20], [21] is used to calculate the correction value. Documents gradually lose their weight (or memory) according to distances.

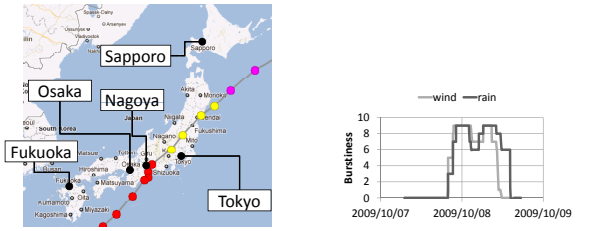
$$\delta(dp, up) = total\_time \times \alpha^r,$$
$$r = \frac{calc\_distance(dp, up) - d_{min}}{d_{max} - d_{min}}, \tag{12}$$

where α is a forgetting factor, $total\_time$ is elapsed time between the start time and current time, d_{max} is the maximum distance between the user and the locations where the documents were created, and d_{min} is the minimum distance between the user and the locations where the documents were created.

5. Experimental Results

To evaluate the location-based burst detection algorithm, we used an actual SDS that is composed of crawling tweets on Twitter about typhoon Melor in 2009. The number of tweets is 504. The time period is from 07:00:11 October 7, 2009 to 13:35:01 October 9, 2009. Typhoon Melor resulted in landfall at the Chita Peninsula in Japan on October 8 after 5 a.m. (JST). Rainfall increased at many places; in particular heavy rains were observed in Osaka, Mie, Tokyo and the Saitama Prefecture. Fig.4(a) shows the path of typhoon Melor.

In the experiments, we select two words; "wind" and "rain," which are the first and the second score in tf*idf results respectively. When the typhoon was on its way toward users, these two words generates the most interest.



(a) Typhoon track. (b) Burst of "wind" and "rain."

Fig. 4: Typhoon Melor.

The five major cities of Japan, Fukuoka, Osaka, Nagoya, Tokyo, and Sapporo are set as the users' positions (Fig. 4(a)). Since the typhoon was headed toward areas near Nagoya, it was a topic concern in the nearby cities of Osaka and Tokyo at that time. The effects of the typhoon began to first appear in Fukuoka because the typhoon went from southwest to northeast. Futhermore, the effects of the typhoon were felt last in Sapporo.

Fig. 4(b) shows the bursts of "wind" and "rain" which are extracted using Kleinberg's burst detection algorithm. Parameter β is set to 1.1 and γ is set to 0.05. The typhoon landed at 5:00 a.m. on October 8 and left the Japanese islands in the evening. Both words are highly bursty between the night of October 7 and the evening of October 8. The degree of burstiness for the word "rain" remained high until the end of the time period. The damage to the Japan islands from the typhoon not only due to wind but also rain. Therefore, concerns about rain continued until the end of the time period, where as concerns about wind decreased earlier in the time period. This result indicates that Kleinberg's burst detection can extract the bursts of words.

Fig. 5 shows the bursts of the word "wind" extracted using the location-based burst detection algorithm. In the graphs, TDC and FFC are the proposed method. TDC indicates that time-difference-based correction is used and FFC indicates that forgetting-factor-based correction is used. CDBD indicates the Distance-Cutoff-based Burst Detection method. In CDBD, the cutoff distance $cutoff$ is set to $150km$. Fig. 5(a), Fig. 5(b) and Fig. 5(c) are the results at Fukuoka. Fig. 5(d), Fig. 5(e) and Fig. 5(f) are the results at Osaka. Fig. 5(g), Fig. 5(h) and Fig. 5(i) are the results at Nagoya. Fig. 5(j), Fig. 5(k) and Fig. 5(l) are the results at Tokyo. Fig. 5(m), Fig. 5(n) and 5(o) are the results at Sapporo. Similarly, Fig. 6 shows the bursts of the word "rain" extracted using the location-based burst detection algorithm.

The graphs shows that the words "wind" and "rain" are bursty in Fukuoka during the initial time period. Then, the degree of burstiness quickly reduces. Since Fukuoka is the most west of five cities, the attention paid to the typhoon had risen there earlier than other locations. Moreover, since the typhoon left far away from Fukuoka, the words "wind" and "rain" became less interesting topics in Fukuoka. Therefore, these results provide accurate information for users in Fukuoka. Similarly, in Osaka, the burst appeared from the time when only a few is late compared with Fukuoka. Since it is closer to the typhoon than Fukuoka,

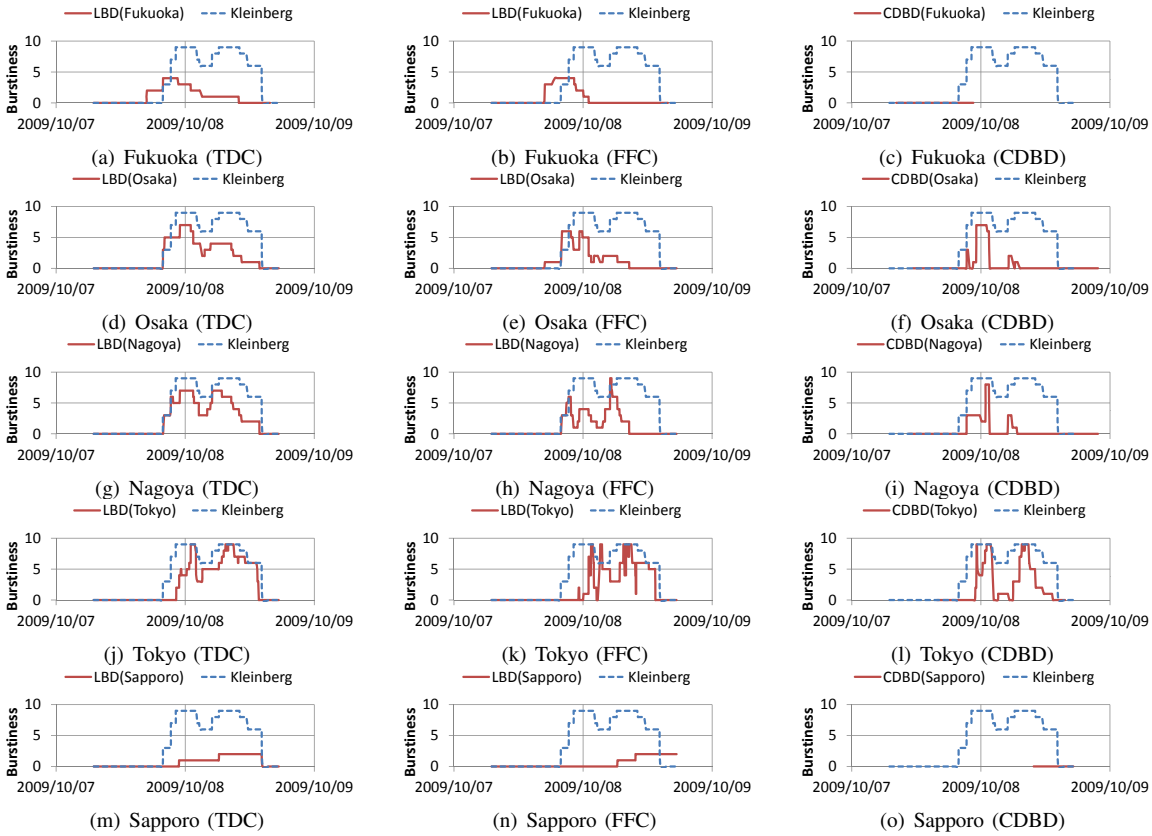


Fig. 5: Results of Word "wind."

the burst state of Osaka continued longer than that of Fukuoka. Nagoya is the closest to where the typhoon resulted in landfall. This resulted in those words being the most bursty around the time of landfall. Tokyo is east of Nagoya. A burst was initiated there slightly after Nagoya. A burst appeared the latest in Sapporo as the typhoon approached it last. Futhermore, the results accurately reflected the typhoon's minimal influence in Sapporo.

On the other hand, CDBD detected the words "wind" and "rain" are not bursty in Fukuoka and Supporo (Fig. 5(c), Fig. 5(o), Fig. 6(c) and Fig. 6(o)). This is because CDBD only considers documents within the cutoff distance $cutoff$. Moreover, in CDBD, burst of "wind" and "rain" appear in Tokyo when the typhoon made landfall. The landfall location is not located within $150km$. Therefore, almost all documents are located in more than $150km$ from Tokyo. This resulted in no bursty appearance in Tokyo at that time.

Fig. 7 shows the results of the word "wind" using CDBD with four different cutoff distances in Tokyo. If the cutoff distance is small, the period of burst is short, whereas, if the cutoff distance is large, the period of burst is long. In CDBD, it is difficult for users to select the best cutoff-distance. The proposed location-based burst detection algorithm dose not need any cutoff-distance. Therefore, our algorithm can detect location-based bursts easier and more correct than CDBD.

6. Conclusion

This study focuses on a document stream that consists of documents containing creation time and location information. We call this type of document stream a spatiotemporal document stream (SDS). We propose a novel algorithm for detecting location-based bursts in SDS. To evaluate the new location-based burst detection algorithm, we use an actual spatiotemporal document stream composed of crawling tweets on Twitter. The experimental results show that the algorithm can detect location-based bursts that vary with user location. In future work, we need more performance evaluations and comparsions with other work.

Acknowledgment

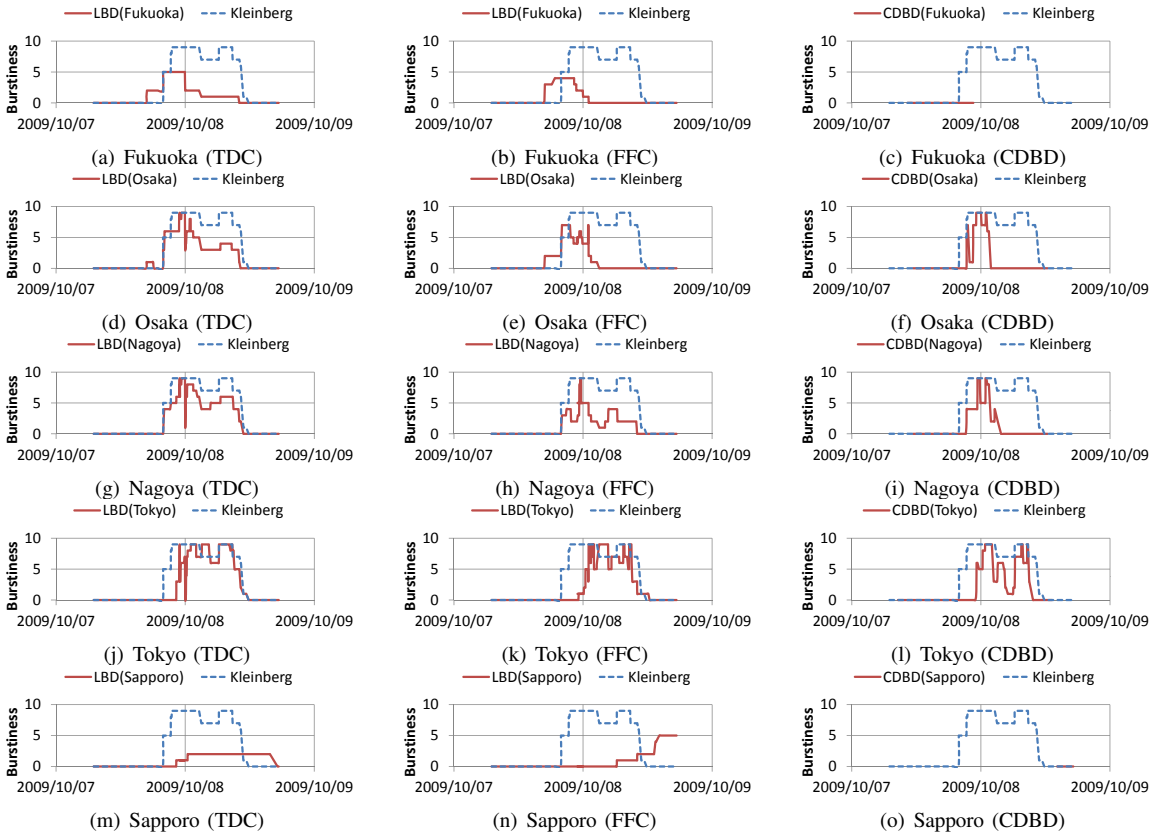
This work was supported in part by a Grant-in-Aid for Young Research (B) (No.23700124) from Ministry of Education, Culture, Sports, Science and Technology in Japan and a Grant-in-Aid for Scientific Research (C) (2) (No.20500137) from the Japanese Society for the Promotion of Science, Japan.

References

[1] M. Ebner and M. Schiefner, "Microblogging - more than fun," in *In Proceedings of the IADIS Mobile Learning Conference 2008*, 2008, pp. 155–159.

[2] J. M. Kleinberg, *Temporal Dynamics of On-Line Information Streams*. Springer, 2006.

[3] J. M. Kleinberg, "Bursty and hierarchical structure in streams," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002, pp. 91–101.



(a) Fukuoka (TDC) (b) Fukuoka (FFC) (c) Fukuoka (CDBD)

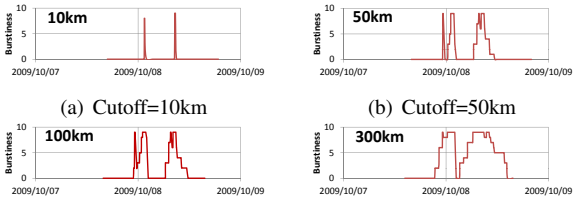
(d) Osaka (TDC) (e) Osaka (FFC) (f) Osaka (CDBD)

(g) Nagoya (TDC) (h) Nagoya (FFC) (i) Nagoya (CDBD)

(j) Tokyo (TDC) (k) Tokyo (FFC) (l) Tokyo (CDBD)

(m) Sapporo (TDC) (n) Sapporo (FFC) (o) Sapporo (CDBD)

Fig. 6: Results of Word "rain."



(a) Cutoff=10km (b) Cutoff=50km

(c) Cutoff=100km (d) Cutoff=300km

Fig. 7: Comparisons of Results of Word "wind" in Tokyo.

[4] J. M. Kleinberg, "Bursty and hierarchical structure in streams," *Data Mining and Knowledge Discovery*, vol. 7, no. 4, pp. 373–397, 2003.

[5] J. Allan, R. Papka, and V. Lavrenko, "On-line new event detection and tracking," in *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 1998, pp. 37–45.

[6] Y. Zhu and D. Shasha, "Efficient elastic burst detection in data streams," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003, pp. 336–345.

[7] X. Zhang and D. Shasha, "Better burst detection," in *Proceedings of the 22nd International Conference on Data Engineering*, 2006, pp. 146–149.

[8] G. P. C. Fung, J. X. Yu, P. S. Yu, and H. Lu, "Parameter free bursty events detection in text streams," in *Proceedings of the 31st international conference on Very large data bases*, 2005, pp. 181–192.

[9] X. Wang, C. Zhai, X. Hu, and R. Sproat, "Mining correlated bursty topic patterns from coordinated text streams," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2007, pp. 784–793.

[10] D. He and D. S. Parker, "Topic dynamics: an alternative model of bursts in streams of topics," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010, pp. 443–452.

[11] R. Kumar, J. Novak, P. Raghavan, and A. Tomkins, "On the bursty evolution of blogspace," in *Proceedings of the 12th international conference on World Wide Web*, 2003, pp. 568–576.

[12] K. K. Mane and K. Börner, "Mapping topics and topic bursts in pnas," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101 Suppl 1, pp. 5287–5290, 2004. [Online]. Available: http://arxiv.org/abs/cs/0402029

[13] J. Yao, B. Cui, Y. Huang, and X. Jin, "Temporal and social context based burst detection from folksonomies," in *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI 2010)*, 2010.

[14] Q. He, K. Chang, E.-P. Lim, and J. Zhang, "Bursty feature representation for clustering text streams," in *Proceedings of the Seventh SIAM International Conference on Data Mining*, 2007.

[15] J. Leskovec, L. Backstrom, and J. Kleinberg, "Meme-tracking and the dynamics of the news cycle," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 497–506.

[16] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: real-time event detection by social sensors," in *Proceedings of the 19th international conference on World wide web*, 2010, pp. 851–860.

[17] Z. Cheng, J. Caverlee, and K. Lee, "You are where you tweet: a content-based approach to geo-locating twitter users," in *Proceedings of the 19th ACM international conference on Information and knowledge management*, 2010, pp. 759–768.

[18] Z. Yin, L. Cao, J. Han, C. Zhai, and T. Huang, "Geographical topic discovery and comparison," in *Proceedings of the 20th international conference on World wide web*, 2011, pp. 247–256.

[19] H. Yang, S. Chen, M. R. Lyu, and I. King, "Location-based topic evolution," in *Proceedings of the 1st international workshop on Mobile location-based service*, 2011, pp. 89–98.

[20] "Online data mining for co-evolving time sequences," in *Proceedings of the 16th International Conference on Data Engineering*, 2000, pp. 13–22.

[21] Y. Ishikawa, Y. Chen, and H. Kitagawa, "An on-line document clustering method based on forgetting factors," in *Proceedings of the 5th European Conference on Research and Advanced Technology for Digital Libraries*, 2001, pp. 325–339.

Definition of Table Similarity for News Article Classification

Taeho Jo
School of Computer and Information Engineering
Inha University
Incheon, South Korea
tjo018@inha.ac.kr

***Abstract*—In this research, we propose method for measuring the similarity between tables. Previously, the table based approach was proposed, but the categorical scores indicating how much the text is relevant to the given category may be overestimated or underestimated by the given text length. As the solution to the problem, in this research, we encode texts into fixed sized tables, define the operation for computing the similarity between two tables as a normalized value, and characterize it mathematically. As the benefits from this research, the categorical score is not influenced by text lengths and the performance is expected to be better and more stable. As the empirical validation, the proposed approach will be compared with the traditional ones with respect to their performance and stability in the test data: 20NewsGroups.**

***Keywords*-News Article Classification, Table Similarity Measure**

I. Introduction

Text categorization is defined the process of assigning one or some of predefined categories to each document. For the task, a list of categories should be predefined and sample documents which are manually labeled by one or some of the categories should be prepared as its preliminary tasks. Techniques of text categorization are necessary for processing and managing efficiently textual data which are growing explosively in information systems; many state of the art approaches have been developed since 1990s. The task is regarded as an instance of pattern classification where each object is classified into its own label.

In order to use a previously developed approach for text categorization, we must encode documents into numerical vectors. Encoding them so causes the two main problems: huge dimensionality and sparse distribution. The first problem, 'huge dimensionality', refers to the phenomena where documents are encoded into too many dimensional numerical vectors for preventing information loss. In spite of using feature selection methods, documents are usually encoded into several hundred dimensional vectors. Under the problem, it takes very much cost for processing each document in terms of time and system resource, and many training examples are required proportionally to the dimension for avoiding over-fitting.

The proposed version is improved over the previous one with three aspects. For first, in the previous version, texts are encoded into variable sized tables, whereas, in this version, they are done into constant sized ones. For second, in the previous version, the categorical scores are computed by summing weights of tables simply, whereas in this version, they are computed by the proposed operation which is characterized mathematically. For third, in the previous version, the categorical scores are only real values, while in the current version, they are given normalized values. Therefore, in this version, we expect more stable performance as well as better performance.

We expect the three benefits from this research. For first, we overcome the overestimation and the underestimation by variable text lengths. For second, the categorical scores are given as normalized values between zero and one independently of domains; the categorical estimations are performed more stably. Compared with traditional approaches, the proposed approach is expected to have its more stable performance over corpus as well as its better performance. Together with the previous version, the proposed version also solves the main problems in encoding texts into numerical vectors.

This article consists of the five sections. In section 2, we will survey the previous research relevant to this research. In section 3, we describe the proposed version of table based matching algorithm in detail. In section 4, we validate empirically the proposed method by comparing it with the popular approaches, considering both performance and stability. In section 5, as the conclusion of this research, we mention the significances and the remaining tasks of this research.

II. Previous Works

This section is concerned with the previous research relevant to this research. Even if various kinds of approaches to text categorization are available, in this research, we count only three typical ones, KNN, Naive Bayes, and Support Vector Machine. In this section, we also survey the previous solutions to the problems in encoding texts into numerical vectors. In spite of its better performance of previous version, we will point out its demerits and mention how to improve it. Therefore, in this section, we will explore the previous research in the three directions.

Let's mention the KNN, the Na?ve Bayes, and the SVM as the three typical approaches to text categorization. The KNN

was used for text categorization by Massand et al and Yang in 1992 and 1999, respectively [1][2]. The Naive Bayes was used by Mladenic and Grobelink and Eyheramendy et al, in 1999 and 2003, respectively [3][4]. The SVM was used for spam mail filtering by Drucker et al [5] and it was mentioned as typical approach to text categorization by Cristianini and Shawe-Taylor [6]. However, it requires to encode texts into numerical vectors for using one of the three approaches for the text categorization.

There were previous attempts to solve the problems in encoding texts into numerical vectors. In 2000, Jo initially encoded texts into string vectors instead of numerical vectors as the alternative representations of texts [7]. In 2002, Lodhi et al proposed the string kernel as a kernel function in using the SVM for the text categorization [8]. In 2007, Lee and K. Kageura tried to solve the problems where many examples are required from the huge dimensionality by generating the virtual documents [9]. The trials show that the problems in encoding texts into numerical vectors were realized.

We started to encode texts into tables instead of numerical vectors and string vectors. In 2008, Jo and Cho created initially the table based matching algorithm as the approach to the text categorization [10]. In same year, Jo applied it to soft text categorization where more than one category may be assigned to each text [11]. In same year, Jo proposed the table based approach to the text clustering as well as the text categorization [12]. The previous version of the table based algorithm solved the problems in encoding texts into numerical vectors, but it has its own demerit where the categorical scores are overestimated or underestimated by variable sized texts.

We need to consider the demerits of the previous version, even if it was applied successfully to text categorization. Even more, the string kernel proposed by Lodhi et al failed to improve the text categorization performance. It is not easy to implement the text categorization algorithms where texts are encoded into string vectors, because operations on string vectors are not defined systematically, mathematically, and theoretically. The previous version of the table based matching algorithm was very unstable because of the bias by text lengths. Therefore, the task of this research is to improve the table based approach into the more stable version.

III. Normalized Table Matching Algorithm

This section describes a table based matching approach to text categorization. Figure 1 illustrates conceptually the architecture of the proposed text categorization system. The part, 'Encoding' encodes a document into a table as the interface of the system, and will be described in detail in section III-A. In section III-B, we will describe the process of computing a similarity between two tables; the computation is used for classification of unseen documents. In section III-C, we will describe the process of learning sample labeled documents and classifying unseen documents using the proposed approach.

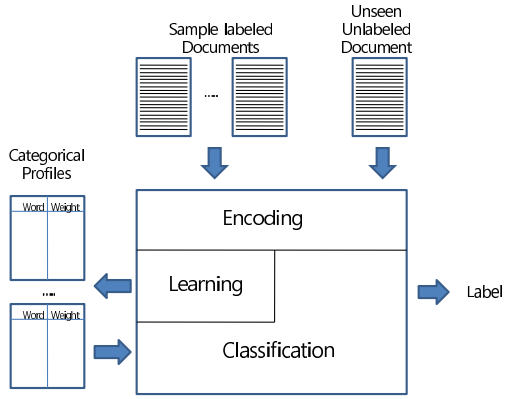


Figure 1. The Process of Indexing Corpus

A. Document Encoding

This section is concerned with the part, 'Encoding' of the architecture of the text categorization system which is illustrated in figure 1. Here, *document encoding* is defined as the process of mapping a document into a table. Figure 2 illustrates the process of encoding a document so through the five steps. As illustrated in figure 2, a particular document is given as the input and its corresponding table is generated as the output. In this section, we will describe in detail each of the five steps involved in the document encoding.

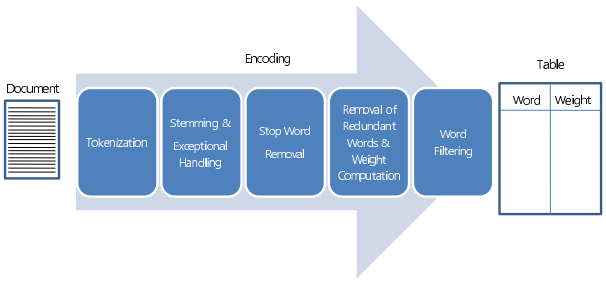


Figure 2. The Process of Encoding a Document into a Table

The first step of document encoding is tokenization as shown in figure2. A full text in a document which is wirtten in a natural language is given as the input of this step. This step, 'tokenization', segments a full text into tokens by white space or punctuation mark. The step generates a list of tokens as the output. A token refers to a word in its raw form.

The second step of document encoding refers to stemming & exception handling. The list of tokens which is generated from the previous step is given as the input of this step. This step converts each token into its root form by stemming it or applying an exception rule to it. This step is carried out by loading stemming & exception rules each of which specifies conversion of each word into its root. Therefore, a list of words in their root forms is generated as the output of this step.

The third step of document encoding is to remove stop words from the list of words. A stop word refers to a grammatical word which do only grammatical functions, irrelevantly to the content of the original document. In English, conjunctions, pronouns, prepositions, and so on belong to this kind of words. Removing the kind of words is necessary for processing documents more efficiently in context of text mining and information retrieval. This step usually remains verbs or nouns as its output.

The fourth step is to remove redundant words and compute weights of each of remaining words. A list of words in their root forms except stop words is given as the input of this step and redundant words are removed among them. The weight of each word indicates how much important it is in terms of the relevancy to the content of the given document. The weight is computed using equation (1),

$$weight_i(w_k) = tf_i(w_k)(\log_2 D - \log_2 df(w_k) + 1) \quad (1)$$

where $weight_i(w_k)$ indicates the weight of word, w_k, relevantly to the content of document, i; $tf_i(w_k)$ indicates the frequency of the word, w_k, in the document, i; D means the total number of documents in the referenced corpus; and $df(w_k)$ indicates the number of documents of the corpus including the word, w_k. A particular corpus is required for computing weights of words using equation (1), and a list of pairs of a word and its weight is generated as the output of this step.

Although stop words and redundant words are removed, we need to filter out additionally words with lower weights for more efficient processing. The previous version which Jo and Cho proposed in 2008 [10], omitted the word filering, so it took very much time for processing documents for tasks of text categorization. Especially when computing a similarity between two tables, its complexity is quadratic $O(n^2)$, so we need to cut down the size of tables as much as possible, minimizing information loss. We can consider two kinds of schemes for filtering words with their lower weights. One is called rank filtering where a fixed number of words with their higher weights is selected, and the other is called threshold filtering where weights of words are normalized as continuous values between zero and one, and words with their weights higher than the threshold are selected.

B. Similarity between two Tables

This section is concerned with the computation of a similarity between two tables. Two tables each of which represents a document or a group of documents are given as the input. A table which consists of words shared by the two tables is derived from the two tables. A similarity between the two tables is computed based on the shared words in the derived table. Whether weights of words are given as normalized or unnormalized values, the similarity is always generated as a normalized value.

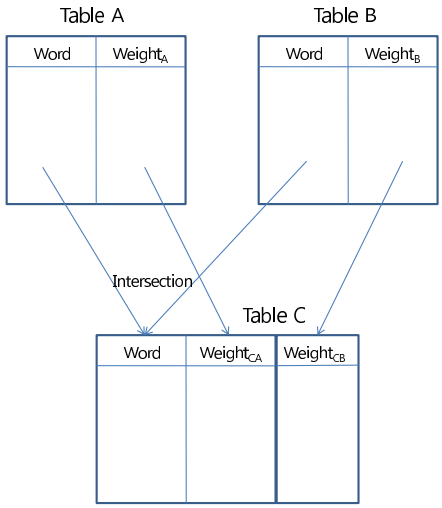


Figure 3. The Process of Deriving a Table from the Two Input Tables

The process of deriving a table from the two input tables is illustrated in figure 3. Let table A and table B in figure 3 be the source tables. Let table C be the destination table which is derived by extracting shared words from the source tables. Table C consists of words shared by both source tables. Each entry of the destination table consists of a shared word and its two weights: one is from table A and the other is from table B.

A similarity between two tables is computed using equation (2)

$$similarity = \frac{weight_{CA} + weight_{CB}}{weight_A + weight_B} \quad (2)$$

where $weight_{CA}$ and $weight_{CB}$ indicate sums of weights of common words from table A and B, respectively, and $weight_A$ and $weight_B$ indicate sums of weights of words in table A and B, respectively. A similarity computed by equation (2) is bound from 0 to one as a normalized value. If there is no shared word between the two tables, the similarity becomes zero. If the two source tables are exactly same as each other, the similarity becomes one. Therefore, even if

weights of words are given as non-normalized values, it is guaranteed that the similarity is given as a normalized value.

We demonstrate the computation of the similarity through a simple example. Two source tables are given in table I. The destination table is derived from the two source tables as illustrated in table II. The similarity between the two source tables is computed based on the destination table using equation (2) as follows:

$$\frac{1.5}{1.2 + 1.7}$$

Therefore, the similarity in this example becomes 0.51.

Table I
TWO SOURCE TABLES: TABLE A (LEFT) AND TABLE B (RIGHT)

Table A		Table B	
computer	0.3	computer	0.6
system	0.2	system	0.4
hardware	0.5	information	0.5
CPU	0.2	data	0.2

Table II
DESTINATION TABLE: TABLE C

computer	0.3	0.6
system	0.2	0.4

C. Learning & Classification

This section is concerned with the process of learning sample labeled documents and classifying an unseen document. There exist two functions in the text categorization system: learning and classification. Learning refers to the process of building rules or equations of classification using sample labeled documents in context of text categorization. Classification refers to the process of classifying an unseen document based on the defined rules or equations. Note that learning is prerequisite for classification.

In the view of the proposed text categorization system, learning is defined as the process of building tables corresponding to categories using sample labeled documents. Categories are predefined, and sample documents are allocated to their corresponding categories. Figure 4 illustrates the part, 'Learning', in the proposed text categorization system which is illustrated in figure 1. From a collection of documents labeled identically, as the learning process, a table is built and called categorical profile in this paper; learning is carried out by attaching the concatenation which concatenates full texts of documents into a full text, to the process of encoding which is illustrated in figure 2. Therefore, learning generates categorical profiles as many as categories as its output as shown in figure 4.

Classification is defined as the process of deciding one of the predefine categories to each unseen document. The process of classifying an unseen document is illustrated in figure 5. An unseen document is encoded into a table by the process illustrated in figure 2, as shown the left part of figure 5. As shown in the middle part of figure 5, similarities of the table with categorical profiles given as tables are computed; the computation was already described in section III. Therefore, the unseen document is classified into the category corresponding to the maximum similarity between its table and the corresponding categorical profile.

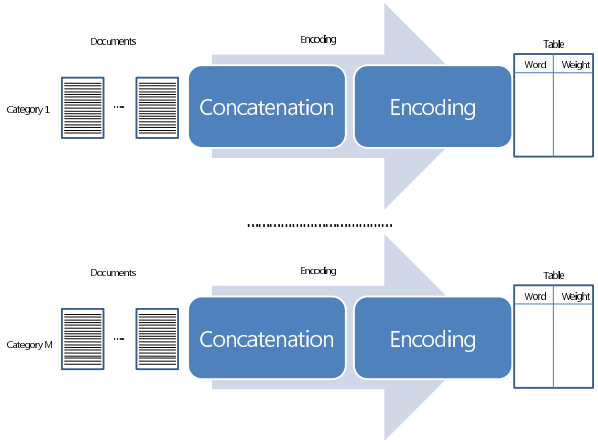


Figure 4. The Process of Learning Sample Labeled Documents in the Proposed Text Categorization System

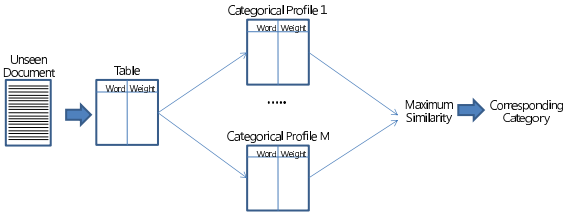


Figure 5. The Process of Classifying a particular Unseen Document

IV. EXPERIMENTS AND RESULTS

This section is concerned with the set of experiments carried on the test data: 20NewsArticles. The test data is larger collection of news articles than NewsPage.com. Like the case in the previous set of experiments, texts are encoded into one hundred dimensional numerical vectors and ten sized tables. The configurations of the approaches and the procedure in this set of experiments are same to those in the previous set of experiments. In this section, we describe the collection of news articles called 20NewsGroups, present the empirical results, and discuss on them.

We use the collection of news articles called '20NewsArticle' as the test collection for evaluating the approaches. It was downloaded from the web site, http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html. In the collection, entirely, 20,000 news articles and 20 categories are available. The 20 categories are given as

the two level hierarchical structure; the first level has the four categories and the second level has the 20 categories; each category in the first level has the five categories in the second level. In this set of experiments, we selected the four categories: computer, record, natural science, and social science.

The configurations of the approaches participating in the experiments are presented in table IV. In using the KNN, the number of nearest neighbors is set three. In using the SVM, the kernel function, the capacity, and the maximum iteration are set the inner product, 4.0, and 1,000, respectively. In using the MLP, the learning rate, the number of hidden nodes, the iterations, are set 0.1, 10, and 1000, respectively. Texts are encoded into 100 dimensional numerical vectors and ten sized table for using the three machine learning algorithms and the proposed approach, respectively.

Table III
THE CONFIGURATIONS OF THE APPROACHES PARTICIPATING IN EXPERIMENTS

Approaches	Configurations	Document Encoding
Naive Bayes	N/A	100 dimensional numerical vectors
KNN	$K = 1, 3$	
NNBP	#Hidden Nodes=10 #Epochs=500 Learning Rate = 0.1 Sigmoid Function	
SVM	Capacity=4 Inner Product	
Proposed Approach	10 sized Tables	

The results from this set of experiments are presented in figure 6. In figure 6, the y-axis indicates the F1 measure of each approach, and the given task is decomposed into the four binary classifications corresponding to the categories. The proposed approach shows its better performance in the three categories among the four categories as shown in figure 6. It shows its comparable performance to the others in the first category, 'comp'. Therefore, we conclude from this set of experiments that the proposed approach works better, generally.

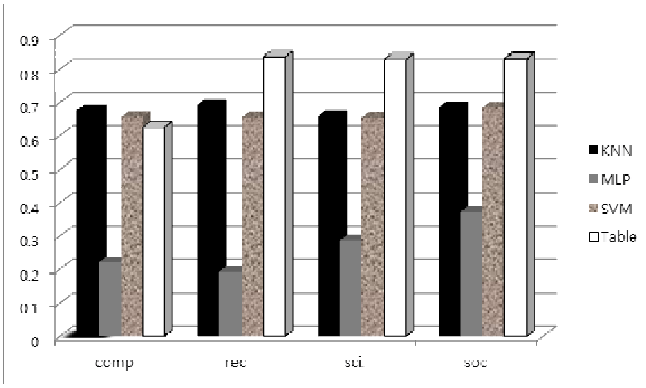


Figure 6. The Results from the Set of Experiments in 20NewsGroups

The overall performances of the four approaches spanning over the four categories are presented in table IV. As shown table 4, the proposed approach has its largest F1 measure of the four approaches. Unlike the previous set of experiments, it has its higher variance; its stability is less than the three approaches. The higher variance comes from the relatively smaller F1 measure in the first category, as shown in figure 6. The KNN and SVM are more stable in this set of experiments.

Table IV
THE OVERALL PERFORMANCE AND STABILITY OF APPROACHES IN 20NEWSGROUP

	KNN	MLP	SVM	Table Matching
F1 Average	0.6774	0.2677	0.6621	0.7801
F1 Variance	0.0001546	0.004852	0.0001520	0.008028

V. CONCLUSION

Let's consider the significances of this research. Like the previous version of the table based algorithm, we are free from the three main problems in encoding texts into numerical vectors: huge dimensionality, sparse distribution, and poor transparency. Because the tables representing texts are symbolic, we trace the classification more easily, in order to provide the evidences. In the proposed version, the categorical scores of the given text are independent of its length. The table based approach is improved to reach more stable performance as shown in the set of experiments presented in section 4.

In order to reinforce the current research, we may consider the four directions of further research. In the first direction, we need to validate the categorization performance of the proposed approach in multiple labels categorization as well as single label one. In the second direction, we may consider that a document or documents are encoded into a committee of tables rather than a table by using multiple schemes for weighting words. In the third direction, in order to keep efficiency and reliability of the proposed approach, we may build the text categorization system in evolutionary fashion by incrementing tables gradually. In the last direction, we may implement a text categorization system as a prototype program where the proposed approach is adopted.

REFERENCES

[1] B. Massand, G. Linoff, and D. Waltz, "Classifying News Stories using Memory based Reasoning", pp59-65, The Proceedings of 15th ACM International Conference on Research and Development in Information Retrieval, 1992.

[2] Y. Yang, "An evaluation of statistical approaches to text categorization", pp67-88, Information Retrieval, Vol 1, No 1-2, 1999.

[3] D. Mladenic and M. Grobelink, "Feature Selection for unbalanced class distribution and Na?ve Bayes", pp256-267, The Proceedings of International Conference on Machine Learning, 1999.

[4] S. Eyheramendy and D. Lewis and D. Madigan, “On the Naive Bayes Model for Text Categorization ”,pp165-171, The Proceedings of the 9th International Workshop on Artificial Intelligence and Statistics, 2003.

[5] H. Drucker, D. Wu, and V. N. Vapnik, “Support Vector Machines for Spam Categorization”, pp1048-1054, IEEE Transaction on Neural Networks, Vol 10, No 5, 1999.

[6] N. Cristianini and J. Shawe-Taylor, “Support Vector Machines and Other Kernel-based Learning Methods, Cambridge University Press, 2000.

[7] T. Jo, “NeuroTextCategorizer: A New Model of Neural Network for Text Categorization”, pp280-285, The Proceedings of ICONIP 2000, 2000.

[8] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, “Text Classification with String Kernels”, pp419-444, Journal of Machine Learning Research, Vol 2, No 2, 2002.

[9] K. Lee and K. Kageura, “Virtual relevant documents in text categorization with support vector machines”, pp902-913, Information Processing and Management, Vol 43, No 4, 2007.

[10] Taeho Jo and Dongho Cho, ”Index Based Approach for Text Categorization”, pp127-132, International Journal of Mathematics and Computers in Simulation, Vol 2, No 1, 2008.

[11] Taeho Jo, ”Table based Matching Algorithm for Soft Categorization of News Articles in Reuter 21578”, pp875-882, Journal of Korea Multimedia Society, Vol 11, No 6, 2008.

[12] Taeho Jo, ”Single Pass Algorithm for Text Clustering by Encoding Documents into Tables”, pp1749-1757, Journal of Korea Multimedia Society, Vol 11, No 12, 2008.

Early Results of Composite NER Algorithm for Résumé Corpora Distillation

Sahil Patwardhan[1], **Pallavi Agarwal** [2], **and Pooja Sunder**[2]
[1]Department of Computer Engineering, College of Engineering Pune, Pune, Maharashtra, India

Abstract - *In this paper we report the results of a comparative study of statistical models, Maximum Entropy (MaxEnt) and Conditional Random Fields (CRF), for the task of Named Entity Recognition (NER) specifically for résumé corpora. Following the first stage, which consisted of the implementation and result analysis of the above mentioned systems, an ensemble method, which combines the results of both models using the established techniques of addition and multiplication, was implemented. Further, we also propose a composite algorithm which, as desired, achieves better precision and recall as compared to the other models mentioned above. We trained the MaxEnt and CRF entirely with an annotated corpus of CVs marked-up with term classes such as Degree, Designation etc., by incorporating code to identify features specific to résumé documents. Using cross-validation technique we achieved an F-score of 87.91%.The paper covers the methodology adopted to carry out our experiment, discusses experimental results and scope for future.*

Keywords: Maximum Entropy, CRF, NER, Data Mining

1 Introduction

Today major companies are faced with an overwhelming number of résumés flowing into their Talent Acquisition Centers. It is a time-consuming task to sift through the hundreds and thousands of résumés and actually be able to differentiate the potential candidates from the ones who are not. Even a trivial sub-task like getting a bird's eye view of the basic details of the candidates, like their names, educational background and past employers becomes non-trivial when there are so many documents to go through. Another constraint is the varied styles and formats in which these résumés are fashioned, which makes it difficult for a cursory glance to get you answers. Hence we see that here is a demand which has not yet been catered to. The task of named entity recognition can be made use of for résumés and the algorithms can be modeled to accommodate for the recognition of particulars of a candidate, like name, location, degree, domain, past employers etc. The purpose of our experiment is, thus, to make the job of résumé data extraction a more dependable task by combining two Named Entity Recognition (hereafter referred to as NER) algorithms, viz. Maximum Entropy and Conditional Random Fields, thereby aiming to create a reliable composite system that could assist businesses and organizations.

The method followed in conducting this experiment initially involved tagging résumés with classes as mentioned in Table 1 for creating the training data set. A list of features, which would assist in sharpening our model to suit the objective of the experiment, was enlisted. The next task was to model these features into the algorithms mentioned above with the help of certain tools. The algorithms of MaxEnt and CRF were trained by feeding them the same training data set of 250 tagged résumés, tested on 200 untagged résumés and meaningful results were obtained. Working towards the designing of the composite deliverable we conceived, the output obtained from the two individual algorithms was combined and results were compiled for analysis, the detailed discussion of which is given in Section 4. Figure 1 describes the basic idea behind the development of the composite that this paper introduces.

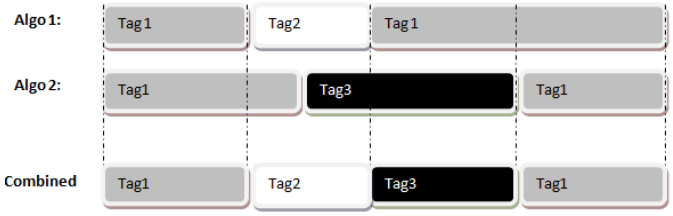


Figure 1. Seed for the Construction of a Composite Algorithm. Algorithm 1 and 2 being Maximum Entropy and Conditional Random Fields (say) respectively, generate tags for all the entities in a text. The combined algorithm does a probalistic analysis of these individual results and gives the entity a final tag. On observing the combined results, the grey areas are the ones where there is an overlap between the tag provided by both the individual algorithms and hence is retained as it is. However in the white and black regions of the combined results, one can observe a conflict between the individual algorithm results which is resolved by the composite algorithm discussed in Section 4 and the final tags are thereby assigned.

2 Algorithms

2.1 Maximum Entropy

Entropy being a measure of uncertainty, Maximum Entropy (hereafter referred to as MaxEnt) is a framework for estimating probability distributions, taking into account the uniformity of the distribution. The classifier is implemented using a regression model that takes into account features of the independent variables and uses them to predict the outcome. The Maximum Entropy principle states that, depending on the prior data, the probability distribution which

best represents the current state of knowledge without assuming anything about that which is unknown, subject to a set of constraints, is the one that is most uniform for the given finite information.[7]

2.2 Conditional Random Fields

Conditional Random Fields (hereafter referred to as CRF) are discriminative undirected probabilistic graphical models. They are used for structured prediction. The model predicts labels taking into account known relationships between observations and thus constructs consistent interpretations. Since it is undirected, the nodes can be divided into exactly two disjoint sets-the observed and output variables and a conditional distribution is then modeled through a fixed set of feature functions.

3 Experiment

3.1 Named Entity Classes

Classes, in the context of résumé database, are the content carrying units within the text.[6] They provide us with the basic details of an applicant, for instance, where he/she has worked previously, which degree he/she holds etc.

The standard entity types found are *Name*, *Location* and *Organization.* However, specific to résumés, finer details of a candidate, such as, work experience, educational background etc. were required to be identified for the purpose of selecting the right candidate for a position in an organization. As a result, the named entity classes given in Table 1 were chosen to represent a candidate and the same were used for annotating the training data résumés.

Table 1. Table of Named Entity Classes

Name	Example	Description
name	Anil Kumar	Name of a person
location	Pune	Location-can be a city, state or country
client	Levi Strauss	Name of the organization his company has catered to
employer	Capgemini Services	Name of the organization the person has worked for
educational_organization	University of California	Name of an educational organization
degree	B.E	Educational qualification
designation	Consultant	Designation held in a company
other	worked for	Other words from the English language

3.2 Features

Features are those formats, styles or particular sequences that help us identify classes from a give text. Features provide evidence that helps to differentiate words belonging to one class from the other thereby causing the entities to be tagged accurately.[10] Moreover it is expected that they will help solve the unknown word problem by finding similarities between tagged words in the training data set and words that have not occurred before. For example, if the AllCaps feature reckons TCS to be an employer class from the training data set then one would expect another similar employer occurring for the first time, say IBM, under the employer class.

To model the tools according to the requirements of the experiment certain features have been added that would pertain to résumé specific content. For example, organization containing a digit-EMC2, institutions containing mixed Caps-CoEP. The table of the features used is given as Table 2.

Table 2. Table of Features Incorporated in the Models

Feature Name	Feature Instance	Applied to Entity
First Word	FW_College	College of Engg. Pune.
Last Word	Pune_LW	College of Engg. Pune.
First two words	FW_College_of	College of Engg. Pune.
Last two words	Engg._Pune_LW	College of Engg. Pune.
Context of previous and next word	Context_at_in	College of Engg. Pune.
Words just before colon in the current line	CLW_Year	Second year
All capital letters along with first letter	Cap_TCS	TCS
Starts with a capital letter, ends with a period	CP_India	in India.
Contains only one capital letter	CO_Bachelor	Bachelor of
All capital letters and period	ACP_Capgemini_Ltd	Capgemini Ltd.
Contains a digit	DIG_EMC2	company EMC2
Mixed Caps	MC_CoEP	CoEP
Dictionary word	-	-
Contains a period	PER_B	B.E

3.3 Data Set

The Data Set consists of both training data as well as testing data. The training set used in the experiment consisted of 250 résumés tagged in XML format for the entities/classes as given in Table 1. The models were tested on a set of 200 résumés.

A certain amount of preprocessing was required before we annotated the training data. All the résumés were in different formats like a Word DOC, PDF. etc. which needed to be converted to standard text format.

The data set consisted of résumés having varied styles of writing - tabular, listed, narrative etc. For training and testing, the résumés were distributed so as to contain a balanced number of both typical and marginal entities.[4]

3.4 Description

The experiment conducted primarily revolves around the task of Named Entity Recognition. There are a good number of algorithms which are capable of performing this task. Specific to our objective of entity recognition in résumés the algorithms chosen are Maximum Entropy and Conditional Random Fields. The data set used for conducting the experiment has been uniform across all the models and so

have the features that have been used to recognize tags. The results obtained from the above mentioned algorithms have provided rather deep insight about their computational efficiency, strengths and weaknesses. The results of the implementations of MaxEnt and CRF have been combined in the ensemble method using addition and multiplication techniques with an aim to obtain better results. Finally, a composite algorithm has been developed to improve both precision and recall parameters.

3.5 Methodology of Implementation

3.5.1 Implementation of MaxEnt

Tool: For the implementation of MaxEnt, the Apache OpenNLP library, a machine learning based toolkit has been used. It supports the most common NLP tasks, such as tokenization, sentence segmentation, part-of-speech tagging, named entity extraction, chunking, parsing and co-reference resolution. This tool has been selected since it meets the criteria for selecting a suitable implementation tool [3] and it is an open source tool whose source code is modifiable.

Pre-processing: The input format for Apache OpenNLP required pre-processing. The training data consisted of a single file which consisted of all the phrases from the résumés which were selected for training along with all the features applicable to the phrase and its class/tag. For the purpose of testing, a file, similar to the one mentioned above, was created except that it did not contain the class/tag of the phrase. Hence, the phrasing logic was very crucial to this model.

Incorporation of Features: Features mentioned in Table 2 were coded at different stages resulting in various revisions of the model.

3.5.2 Implementation of CRF

Tool: For the purpose of implementing a CRF parser, the Stanford NER CRF parser has been used. It is a Conditional Random Field sequence model, along with well-engineered features for Named Entity Recognition in English. This tool again met the criteria[3] and also is an open source tool.

Pre-processing: The Stanford NER CRF parser required the input to be tokenized using a tokenizer, as a result of which each word from every résumé, along with its tag appears on a separate line, and this file is fed as input for training. The same tokenizer is run on the test data without the class appearing alongside each word.

Incorporation of Features: The Stanford NER CRF parser has a provision to recognize a few basic features. Apart from these, feasible features from Table 2 were modeled into the parser.

3.5.3 Ensemble Method

There are a number of techniques which can be used to combine the results. We have adopted two of the established techniques – Union (OR-ing or addition) and Intersection (AND-ing or multiplication) of the results obtained from MaxEnt and CRF individually. Union improves the recall where as intersection improves precision.

3.5.4 The Composite Algorithm

The ensemble methods improve either precision or recall parameters while the other parameter suffers. Through the composite that we propose, whose detailed discussion is provided in Section 4, we aim at improving both these measures of efficiency by performing a probabilistic analysis of the results obtained by both MaxEnt and CRF and resolution of conflict in case the same entity gets tagged differently by the two implementations.

3.6 Results

Results have been calculated in terms of precision, recall and F-score. Precision is the fraction of retrieved instances that are relevant[1]. Precision tells us whether the entities that have been tagged in the test data have been tagged accurately. Recall is the fraction of relevant instances that are retrieved[1]. It is a measure that tells the number of entities accurately tagged in the test data as compared to those tagged.

There is another measure which combines precision and recall, called the F-score. It is a tool to measure the accuracy of testing and is defined as:

$$F = 2\frac{precision.recall}{precision+recall} \quad (1)$$

Table 3. Table of Results

Algorithm		Precision	Recall	F-Score
Maximum Entropy		82.75	60.75	70.07
CRF		90.75	75.0	82.12
Ensemble	AND-ing	92.01	54.25	68.25
	OR-ing	89.12	79.75	84.17

It is observed that that CRF results look better than MaxEnt results[9]. Table 3 also shows the results obtained on combining results from the two individual algorithms. The results of AND-ing and Or-ing, as observed improve precision and recall respectively. Or-ing provides fairly satisfactory results.

4 Details of the Composite Algorithm

4.1 Mathematical approach

The composite algorithm can be formulated mathematically as follows:

$$P(A^*) = \lambda_1 P(A) + \lambda_2 P(C|S) + \lambda_3 P(C|C^*) \quad (2)$$

Where,

$P(A^*)$ = the new probability that will be used for comparison

$P(A)$ = the calculated probability by individual algorithms

$P(C|S)$ = a probability depending on the number of occurrences of the class C in the training data set S. The

probability would be more if the class *C* has occurred more frequently in the training data set.

P(C|C)* = probability calculated depending on the global features. The global features would comprise of events from both the training and testing data set. *C* is the class with the highest probability as calculated by the individual algorithm. If an occurrence of the word has already appeared in the training data set then its class would be included in the set *C**. Similarly, if the entity has occurred in the data set tested till this point, its class is included in *C**. The probability would then be calculated by finding the ratio of the number of occurrences of the class *C* to the cardinality of the multiset *C**.

λ = constant, which in our current system, is set by hand where, $\lambda_1 > \lambda_2 > \lambda_3$.

4.1.1 Justification

The following is a mathematical justification for the approach mentioned in Section 4.3. Let $\lambda_1, \lambda_2 \ldots \lambda_k$ be the constants such that,

$$\sum_{i=1}^{k} \lambda i = 1 \qquad (3)$$

Let $P_1, P_2 \ldots P_k$ be probabilities which adhere to all the properties of a probability. Hence we know that $0 \leq Pk \leq 1$.

$$P = \lambda_1 P_1 + \lambda_2 P_2 + \ldots + \lambda_k P_k \qquad (4)$$

Equation (4) is similar to equation (2) in Section 4.3. Now using the property $0 \leq P_k \leq 1$, we can therefore say that,

$0 \leq \lambda_1 P_1 + \lambda_2 P_2 + \ldots + \lambda_k P_k \leq \lambda_1 . 1 + \lambda_2 . 1 + \ldots + \lambda_k . 1$

$0 \leq \lambda_1 P_1 + \lambda_2 P_2 + \ldots + \lambda_k P_k \leq \lambda_1 + \lambda_2 + \ldots + \lambda_k$

Now using equation (3) we get,

$0 \leq \lambda_1 P_1 + \lambda_2 P_2 + \ldots + \lambda_k P_k \leq 1$

Again, using equation (4) we get,

$0 \leq P \leq 1 \qquad (5)$

Hence the quantity *P* satisfies the property of a probability and justifies equation (2).

4.2 An Example

Word1

MaxEnt:<name>[0.1323] <employer>[0.2614]
<client>[0.2473] <degree>[0.1473] <location>[0.2117]

CRF:<name>[0.1075] <employer>[0.4102]
<client>[0.1502] <degree>[0.1197] <location>[0.2124]

Word2

MaxEnt:<name>[0.2955] <employer>[0.1185]
<client>[0.2575] <degree>[0.1821] <location>[0.1702]

CRF:<name>[0.1956] <employer>[0.0795]
<client>[0.2777] <degree>[0.1517] <location>[0.2955]

Figure 2. Probabilities Obtained from Individual Algorithms for Particular Word Instances

The composite algorithm devised takes the probabilities for each word, for each algorithm and computes a maximum as given by the mathematical formula. For instance, consider probabilities for Word 1 in Figure 2. The maximum observed probability for MaxEnt is 0.2614 while that for CRF is 0.4102. Since both the algorithms have tagged the entity as '*employer*', according to equation (2), the final comparison would be among the same classes and hence, it is futile to run the conflict-resolution algorithm as a result of which the entity is assigned the tag '*employer*'.

However if the two algorithms assign different classes to the same entity, the parameters, *P (C|S)* and *P(C|C*)* as mentioned in equation (2) are computed and the maximum of the resulting probability for both the algorithms are compared. The class with the higher probability is then assigned to the entity.

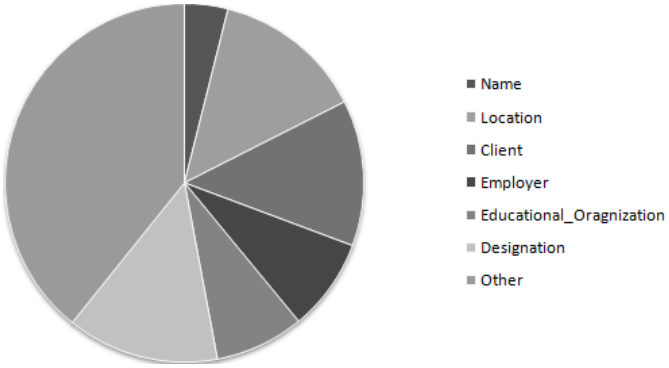


Figure 3. Pie chart showing the distribution of entity classes in the entire data set. This helps us find the quantity *P (C|S)*.

4.3 Result Analysis of Composite Algorithm

Table 4. Table of Results Obtained for Different λ Values for the Composite Algorithm

Parameter values	Precision	Recall	F-Score
λ_1=0.6, λ_2=0.25,λ_3= 0.15	94.34	82.3	87.91
λ_1=0.6, λ_2 =0.3, λ_3 = 0.1	92.14	79.98	85.63

From the results shown in the Table 4 it is clear that, the data set being constant, a change in the parameter values for the combined algorithm, affects the results. It has been observed in our experiment, that the smaller the difference between the values of λ_2 and λ_3, the better is the result that is obtained.

5 Stage-wise Results and Error Analysis

5.1 Stage-wise Revision

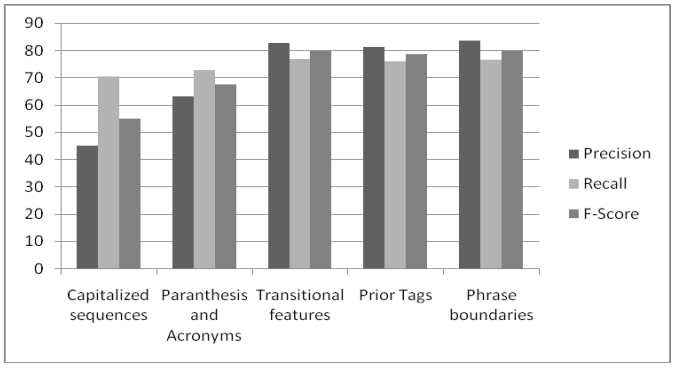


Figure 4. Bar graph showing the improvement in the precision, recall and F-score parameters after every stage of revision for MaxEnt implementation.Transitional features are context words like *in* and *as* in a phrase like '*...worked in IBM as an Analyst.*'while Prior Tags in the same example would be the class *employer* assigned to *IBM* which would help identify *Analyst* as *designation*.

The analysis of the errors encountered at every stage revealed more features that could be incorporated into the algorithms. The stage-wise changes in the results are shown in Figure 4 and 5. It was seen that when handling of parenthesis and acronyms was done for MaxEnt, the F-Score increased by 12.59%. Also, in the case of CRF, on increasing the n-gram length, the F-Score improved by 9.058%. As more features were added to the algorithms, the results obtained were an improvement over the previous results.

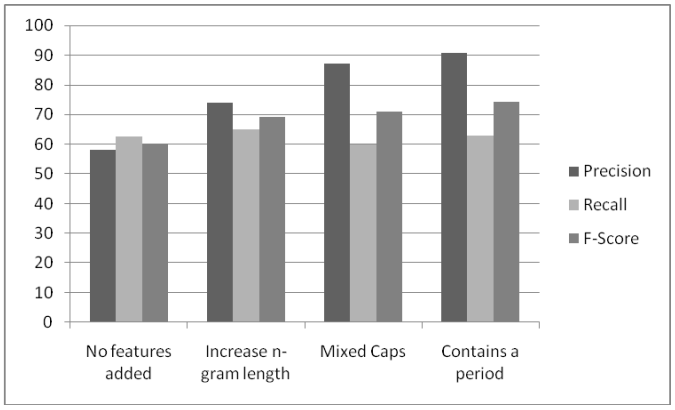


Figure 5. Bar graph showing the improvement in the precision, recall and F-score parameters after every stage of revision for CRF implementation

5.2 Result Analysis

The results obtained after 5-fold cross validation are listed in the Table 5.

Table 5. Table of Results for Specific Entity Classes

Algorithm	Term	Emplo-yer	Degree	Designation	Educational Organization
MaxEnt	Precision	77.0	86.0	94.0	74.0
	Recall	60.0	56.0	92.0	35.0
	F-Score	67.44	67.83	92.98	47.52
CRF	Precision	98.0	94.0	89.0	94.0
	Recall	78.0	67.0	73.0	82.0
	F-Score	86.86	78.23	80.20	87.59

Table 5 shows the measure of efficiency parameters for both algorithms taken for four selected classes which are highly significant in the context of résumés. Results show that CRF performs better in case of EMPLOYER and EDUCATIONAL_ORGANIZATION, while MaxEnt performs better in case of DEGREE and DESIGNATION. In general precision is high in CRF while recall is low.[2] In the case of MaxEnt, it is the other way around. A possible reason for the low precision and recall values for EDUCATIONAL_ORGANIZATION could be the use of abbreviations which make them look like companies. For example, if there are strings such as "IIT, Delhi" and "IBM, India" in the training data, then a phrase like "BITS, Pilani" is not correctly identified.

Results improve greatly if we do not consider named entity boundaries (F-score goes up by minimum 6 to 7% for each class). This is because résumés are generally semi-structured data and the actual relevant information is very sparse in a résumé which is generally very big. State-of-the art NER systems itself are about 90 to 95% accurate. So as of now, an NER application can be used in a 2 step process where NEs (Named Entities) are automatically generated by a classifier in the 1st step and corrections can be made manually which would justify relaxing the tight constraint of considering boundaries.

It is seen that recall in CRF is generally low and the reason for this is inability to capture long range features/context. It models localized features very well and as a result the precision is very good. So work can be done to increase recall of NER by using CRF in a 2 step phase and using long range features in during the 2nd phase.[8] The procedure to be followed is: Train CRF by using normal features, predict on the entire data set (train + test). This is the end of the 1st phase. For the 2nd phase, use the predicted outcomes of training dataset for training and make use of non-local/global/long range features. Predict using this 2nd CRF trained model on the test data. This thereby increases recall by 12 - 15 %.

5.3 Instance of Error Analysis

Due to various limitations, a number of errors were encountered which were basically in the form of false positives and false negatives, both of which are a consequence of incorrect tagging of test data. False positives would be those cases when an entity has been tagged with a particular class where it does not belong to that class and false negative would be a case when a particular entity has not been tagged with its expected class when it should have been. Hence, it

would be easy to understand that the false positive of one class is the false negative of the other.

Considered below, are a few examples to analyze the reasons for errors that occur.

Table 6. Table of Error Instances

Case #	Instance of Error	Observed Tag	Expected Tag
1	BO XI. R2	<degree>	<other>
2	Certified Sr. Project	<educational_organization>	<other>
3	Strauss Signature	<name>	<organization>
4	Passed X	<degree>	-
5	July	<location>	<other>
6	MIS	<employer>	<other>

- Case#1 has been tagged as degree due to its similarity with the format in which degrees are written, M.Sc.
- Case#2 has been tagged as organization due to its resemblance to college names, XYZ Jr. College.
- Case#3 has been tagged as name due the occurrence of Levi Strauss as the owner of the company.
- Case#4 has been tagged as Degree where only 'X' should have been tagged as degree. This is due to the previous word starting with Caps
- Case#5 has been tagged as location due to the occurrence of an instance like Bangalore, India in the training set and Bangalore, July 2008 in the test data.
- Case#6 has been tagged as employer due to similarity with abbreviations like IBM and TCS.

Table 6 above, represents an instance of error analysis performed after obtaining the results of a particular iterative stage (as give in Section 5.1). A similar error analysis was performed after every stage and the model was revised to incorporate more features which could combat the anomalies identified as a part of the analysis.

6 Major Obstacles

6.1 Problems

One of the major problems confronted was sparseness of training data. The training data sets need to be manually tagged, which is a time consuming task. With lack of enough training data little improvement can be expected in case of self learning machines.

Another problem is difficulty in dealing with false positives and false negatives. Their error analysis and hence corrections become cumbersome due to overlapping of certain features and delimiters.

The main problem is identifying phrases (group of words) which are potential named entities and the logic that is used to generate these has a lot of impact on the actual results. We experimented with the task of only classification of phrases (correct phrases were given as input to the classifier) and the results were very impressive. We got minimum 85% F-score for each class which is state-of-the-art results. This shows that if we can identify potential named entities, then we can classify them very well too. Even phrasing logic was improved which lead to even better results.

6.2 Challenges

The most significant challenge identified is the varied styles in which résumés are written and formatted. They are not similar to normal text and do not have a standardized format.

For MaxEnt: Due to the use of point classification, recall will depend upon the number of phrases identified correctly and this in turn will depend on the phrasing logic used for selecting the phrases that will later be classified into one of the classes. Simple capitalized sequences do not give good results since the phrase boundaries are not identified correctly. Addition of some more sophisticated logic like setting a phrase length, adding some exception words, etc. was done.

For CRF: Implementation using Stanford CRF NER reduces complexity to a great extent, except that it is very important to select features which represent training data accurately. Also a large training data is needed to make a good model. [4]

A few other challenges are related to writing style. When the text contains a sentence like *Bush* was elected …, it being the beginning of the sentence makes it difficult to say for sure that it is the occurrence of the name class. The whole text needs to be scanned to find a repetition of the word *Bush*. Similarly the phrase *McDonald* can be Name class or Employer class. For this purpose using global information and not simply phrases can help remove anomalies.

7 Scope for Future

During the course of the experiment we identified areas where changes and corrections can bring about significant improvement in statistics. One such way could be changing of n-gram length. It is believed that the precision and recall values should improve on increasing the n-gram length due to more contexts around a phrase being captured and analyzed, which is reflected in the results obtained though in small measure. [4]

Also implementation of Hidden Markov Model followed by assimilation of its results for the combing strategy may bring about better understanding of the behavior of the models and substantial increase in the F-score.

Use of global information[5], more sophisticated phrasing logic and better design for a composite algorithm are some of the catalysts in the generation of desired level of output.

The concept presented in this paper can be extended to global recruitment, wherein the organization needs to align recruitment to demographics. The model can be refined or expanded to enable building leadership pipeline, leading to

succession planning. For finally, it's all about the right man in the right position!

8 Conclusions

The algorithms that have been used have proved their worth in various fields like Part of Speech tagging, gene name extraction, protein modeling, web-enhanced lexicons etc. Applying them for Named Entity Recognition for résumés is particularly challenging due to the absence of any standardization found in the way résumés are formatted. Also little linguistic help can come to our rescue.

In spite of that, the algorithms have shown satisfactory results and good predictability which increases the scope for further probing and fine-tuning. The features that have been specially incorporated in the system have led to good precision. Certain classes like organization, location and degree have shown fairly good results individually due to qualifiers like *in* and *with* along with ideal features, whereas the results for name badly suffer due to its independent occurrence in the text and the models finding it difficult to identify non-English names .

However, there are limitations around which we need to work our way. While the MaxEnt model, being a point classifier, requires better phrasing logic, the CRF requires a larger training data set. Some post – processing techniques are also being studied. The models are also not very time efficient at this stage but attacking the problems with a Pareto analysis will in the long run improve performance. We also hope that a composite algorithm will greatly help in making results more reliable and useful.

9 Acknowledgement

This paper would not have been possible without the selfless contribution of a number of people, who helped us with their expertise and guidance. We express our gratitude to Tata Consultancy Services (TCS) under whose aegis we pursue our project and our project guides from TCS, Mr. Rajiv Srivastava and Mr. Sachin Pawar who always despite their busy schedule obliged us with their time and suggestions. We are also thankful to our college, College of Engineering, Pune for its amazing infrastructure and resources and our in-house guide Mr. S.P.Gosavi for his encouragement. And last but definitely not the least, friends and family, whose constant support and unconditional help never ceases to amaze us.

10 References

[1] Olson, David L.; and Delen, Dursun (2008); *Advanced Data Mining Techniques*, Springer, 1st edition (February 1, 2008), page 138, ISBN 3540769161.

[2] Liu, Yang / Shriberg, Elizabeth / Stolcke, Andreas / Harper, Mary (2005): "*Comparing HMM, maximum entropy, and conditional random fields for disfluency detection*", In INTERSPEECH-2005, 3313-3316.

[3] Mónica Marrero, Sonia Sánchez-Cuadrado, Jorge Morato Lara, and George Andreadakis (2009) Evaluation of Named Entity Extraction Systems In: Advances in Computational Linguistics. Research in Computing Science, Vol. 41 (2009) , p. 47-58. http://site.cicling.org/2009/RCS-41/047-058.pdf

[4] Stanley F. Chen and Joshua T. Goodman. *An Empirical Study of Smoothing Techniques for Language Modeling*. Technical Report TR-10-98, Computer Science Group, Harvard University, 1998.

[5] Hai Leong Chieu and Hwee Tou Ng, *Named Entity Recognition with a Maximum Entropy Approach*. In: *Proceedings of CoNLL-2003*, Edmonton, Canada, 2003, pp. 160-163.

[6] Nigel Collier, Chikashi Nobata and Jun-ichi Tsujii(2000) : *Extracting the Names of Gene and Gene Products with a Hidden Markov Model*. Int. Conf. Comput. Linguistics, 18, 201-207.

[7] Adam L. Berger, Vincent J. Della Pietra,Stephen A. Della Pietra. *A maximum entropy approach to natural language processing*. Published in: Computational Linguistics Volume 22 Issue 1, March 1996 Pages 39-71

[8] Vijay Krishnan Stanford University, Stanford, CA Christopher D. Manning. *An effective two-stage model for exploiting non-local dependencies in named entity recognition.* Published in: ACL-44 Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics Pages 1121-1128

[9] Dingcheng Li, Karin Kipper-Schuler, Guergana Savova. *Conditional random fields and support vector machines for disorder named entity recognition in clinical texts*. Published in:BioNLP '08 Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing. ISBN: 978-1-932432-11-4

[10] Burr Settles. *Biomedical named entity recognition using conditional random fields and rich feature sets*. Published in: JNLPBA '04 Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications Pages 104-107

[11] Xinnian Mao, Wei Xu, Yuan Dong, Saike He, Haila Wang. *Using Non-Local Features to Improve Named Entity Recognition Recall.* The 21st Pacific Asia Conference on Language, Information and Computation : Proceedings, Vol. 21 (2007) Key: citeulike:9789073

A Content Analysis of Online News Media Reporting on American Health Care Reform

Ahmed YoussefAgha, PhD
Dept. of Applied Health Science
Indiana University
Bloomington, Indiana, USA
E-mail: ahmyouss@indiana.edu

Wasantha Jayawardene, MD
Dept. of Applied Health Science
Indiana University
Bloomington, Indiana, USA
E-mail: wajayawa@indiana.edu

Samuel Obeng, PhD
African Studies Program
Indiana University
Bloomington, Indiana, USA
E-mail: sobeng@indiana.edu

David Lohrmann, PhD
Dept. of Applied Health Science
Indiana University
Bloomington, Indiana, USA
E-mail: dlohrman@indiana.edu

Abstract: **Health Care Reform (HCR) is currently a major concern of many Americans. There are three dimensions that the White House outlined as salient issues, but are also reflective of the fears of the American public. The first dimension, stability and security, refers to the increased dependability of health care and reduction of discrimination against people with health conditions. Dimension two refers to Americans who do not have health insurance and the quality and affordable choices they will gain under the new reform. Dimension three refers to funding concerns and how this new reform will be fiscally managed. Currently, the area of sentiment analysis (SA) has been witnessing a flurry of novel research. However, only a few attempts have been made to build SA for the health domain. In the current study, we report efforts to partially bridge this gap; we used sentiment analysis to analyze the article contents in relation to the three dimensions of the plan. It includes a new annotation scheme that incorporates sentiment analysis of online media that reflected public concerns about the proposed HCR. Chi Square Statistics used for categorical data comparison on the perspectives of the Media data by the plan dimensions before and after the ACA of the proposed HRC. This work will provide more information about the community vision of HCR.**

Keywords: Healthcare reform, Sentiment analysis, Media data

I. INTRODUCTION

A. Significance of Healthcare Reform

Goals of improving the efficiency, restraining expenses, and increasing quality in healthcare have become prominent issues in the health care system, which has prompted the application of business practices to medicine. Average health insurance premiums and individual contributions for family coverage have increased approximately 120% during last 10 years [1]. Health care expenditures in the United States are four times greater than national defense, despite the wars in Iraq and Afghanistan. The U.S. health care system has been blamed for inefficiencies, excessive administrative expenses, inflated prices, inappropriate waste, fraud and abuse. While many people lack health insurance, others who have insurance allegedly receive care ranging from high to inexcusably poor quality. President Obama is focused on creating a national health care system to address some of the current problems.

In criticism of health care in the United States with a focus on saving money, many methodologists, policy makers, and the public seem to dismiss the major disadvantages of other national health care systems and the previous experiences of health care reform in the United States. The Obama administration has high expectations for health care reform in hopes of moving the U.S. toward universal health insurance. Moreover, health care a central part of the Obama domestic agenda, with spending and investments in Children's Health Insurance Program, American Recovery and Reinvestment Act of 2009, and proposed 2010 budget. Many of the groups long opposed to reform are still fighting to derail this agenda. Fears surrounding the rejection of national health insurance include apprehension about increased taxes because of increased numbers of people receiving social services and welfare. Others are worried that they may lose the health insurance they currently pay for.

Those in support of the measure are eager to hear about the options this new health care system would provide for them. Without health insurance, many people do not receive preventative care, and are afraid to see a medical professional when they are sick or injured. As a result, they often suffer and wait until it is absolutely vital they seek help, but that trip to the hospital may cost them an exorbitant about of money.

The American public considers health care reform a critical issue. Since 1992, the public has considered this to be reform one of the four most important problems facing the country [2]. Analysis of public opinion polls shows that the majority of Americans are satisfied with the quality of their health care, but are not satisfied with the cost [2, 3]. Although the majority of Americans agree that reforms are needed to control the costs of health care, there is mass controversy over how these reforms should be realized [4-6]. Blendon & Benson [3] suggest that the inability of the public to agree on a strategy of health care reform has significantly inhibited the political progress of the current reform bill.

Besides rising premiums, another health care issue posing a problem to politicians, the health care professionals, and families, as of 2002, was the estimated 41 million Americans (13.3% of the American population) who did not have health insurance. As many as 80% of uninsured individuals are from working class families, thus without insurance, health care poses a significant financial burden capable of leading to bankruptcy and poor health outcomes.

To address the issue of 41 million Americans not having health insurance, Congress passed the Affordable Care Act (ACA) in March, 2010. The goal of the ACA is to have every American citizen covered by health insurance by 2014 [7]. However, the ACA has faced significant challenges in Congress and caused rigour debate among the American public. Because of this controversy, the ACA was repeatedly amended in order to be passed by Congress. Some suggest that these amendments have significantly "watered down" the ACA and its ability to effectively manage the systemic problems of health care insurance system.

B. Explosion of the Use of the Web

The last few years have been witnessing an explosion of the use of the Web. Many governments, agencies, and administrative bodies around the world are becoming more and more interested in reaching out to citizens using various Web platforms, including social network sites (e.g., Facebook), micro-blogging services (e.g., Twitter), and video-sharing sites (e.g., YouTube). The Obama administration has been remarkably active in communicating with Web users with regard to multiple issues, including ones in the health domain. The opinions of Web users toward Obama's health policies, however, remain buried in the cyberspace. For instance, we know of no efforts to mine Web users' opinions about Obama health care reform plan (OHCRP). More generally, there seems to be little efforts made so far to mine Web data from the health domain.

Analyzing subjective language is a task that belongs to the wider area of subjectivity and sentiment analysis (SSA). SSA is an area that has recently been witnessing a buzzing research interest. Subjectivity analysis aims at sorting out objective (i.e., factual) from subjective (non-factual) information. Non-factual information can be positive, negative, mixed, or neutral and the task of classifying data according to these dimensions is referred to as sentiment analysis.

C. Related Work

Wiebe et al. [8] attempt to classify news data for subjectivity, at the sentence level. Manually annotated a corpus of 1,001 sentences of the Wall Street Journal Treebank Corpus [9] with subjectivity classifications by instructing three humans to assign a subjective or objective label to each sentence. They used five POS features, two lexical features, and a paragraph feature and performed 10-fold cross validation. They report 72.17% accuracy, which is more than 20% points higher than a baseline accuracy obtained by always choosing the majority class.

Bruce & Wiebe [10] performed a statistical analysis of the assigned classifications in the corpus reported in [8]. The analysis showed that adjectives are statistically significantly and positively correlated with subjective sentences in the corpus on the basis of the log-likelihood ratio test statistic G2. Authors found that probability that a sentence is subjective, simply given that there is at least one adjective in the sentence, was 55.8%, even though there were more objective than subjective sentences in the corpus.

Wiebe [11] used manually-identified subjective elements to seed the distributional similarity of such elements. Riloff et al. [12] explored the idea of using subjectivity analysis to improve the precision of information extraction systems.

D. Purpose of the Study

Due to the increasingly urgent need for political progress on health care reform, this study aims to conduct an exploratory analysis of the quality of information available to the public about American health care reform. We were particularly interested in investigating the quality of information available because the OECD indicates that unbiased, fact-based information readily available to the public is one of the four components of successful health care reform [13]. We are focusing on online news articles specifically because, as of 2010, as many Americans get their news from the Internet as they do from TV. Note also that online news articles are easier to analyze as they continue to exist in text form and are easy to access in a timely manner, whereas TV news is difficult because it is fleeting and does not have text attached.

To observe the quality of information found in online news media, we measured the amount of fact-based (objective) information utilized in the news articles, and how often the articles addressed points/issues about health care reform outlined by President Obama in his explanation of the Affordable Care Act [7]. By conducting the analysis, we are investigating the nature, extent, and/or quality information about health care reform the American public was presented by the online media with the intent to propose recommendations for improving the American public's awareness of the

proposed health care program as well as improving the quality of information about health care reform.

II. METHODS

A. Selection of Online News Articles

We selected one hundred and five online news articles regarding health reform from the following sources: ABC News, The Associated Press, Belfast Telegraph, Blogger, Bloomberg Business Week, CBC, CBS, Chicago Tribune, CNBC, CNN, Daily Finance, FOX News, The Guardian, Globe, Huffington Post, Human Events, Las Vegas Review Journal, Life News, Los Angeles Times, Mediaite.com, Medical News Today, MSNBC, New York Times, Newsweek, Politco, Salon.com, Suite 101, The Associated Press, The Australian, The Baltimore Sun, The Fiscal Times, The Irish Times, The New Republic, U.S. News, Virginian-Pilot, Washington Post, WebMD News, and Zap2it. A full list of articles included in the analysis can be found in the Appendix.

Articles were selected from the websites of the sources using the search terms "Obamacare", "Obama health reform", and "health care reform." In the search engine, articles were ordered "by date" in order to identify articles written before and after the ACA was passed. Articles were included if they were written by media professionals who refer to politicians and health care professionals to support their arguments. Fifty-one (51) articles were written before the Affordable Care Act was passed, and fifty-four (54) articles were written after it was passed. The articles were published between October, 2008 and September, 2010.

B. Method of Classification

We conducted a content analysis by extracting paragraphs from the articles. Paragraphs were selected from articles by utilizing Microsoft Word's "Insert -> Table -> Convert Text to Table" function with the separation criteria set at "Paragraphs". A total of 1865 paragraphs were annotated and included in the analysis: 1018 paragraphs before ACA was passed and 847 after ACA was passed.

C. Classification of health care reform dimensions.

The three dimensions of health care reform are *Stability and Security*, *Quality and Affordability*, and *Funding*. A paragraph was classified as "Other" if it did not address any of the above-mentioned dimensions. The criteria for these dimensions were taken from the White House Government Website and a YouTube video of President Obama explaining the details of the health care reform plan (Health Reform in Action, 2010). Examples of paragraphs from the dataset that mention the three dimensions can also be found in Tables (1) and (2). A full explanation of the classification criteria for the three dimensions can be found in Table (3).

D. Classification of subjectivity and objectivity.

A paragraph was categorized as "Subjective" if the author was including personal feelings, opinions, or speculations on a topic. For example, a paragraph that included language such as, "I think this plan is a good idea" or "I believe that health care reform will not benefit Americans" would be classified as subjective. Subjective means author opinion or a combination of opinion quotes by others and authors opinions (editorial).

A paragraph was classified as "Objective" if the author was stating a fact, and not including a subjective judgment (that is, personal feelings, opinions, or speculations on a topic). Thus, a paragraph that included statements such as, "40 million Americans do not have health insurance" without any commentary from the author was classified as an objective paragraph. Objective means all facts (No opinion); it contains few quotes and quotes contain facts, or (quote true experts).

A paragraph was classified as "Objective with Subjective Content" if the author reports (quotes) statements by others that are opinions; it is not the author opinion.

Examples of these paragraphs from the dataset can be found in Tables (1) and (2) below.

Table 1: Examples of Subjective Paragraphs by Dimension

Stability and Security
"For the 85 percent of Americans who already have health insurance, the Obama health plan is bad news. It means higher taxes, less health care and no protection if they lose their current insurance because of unemployment or early retirement" [14].
Quality and Affordability
"Unpopular insurance company practices such as denying coverage to people with existing health conditions would be banned. Uninsured or self-employed Americans would have a new way to buy health insurance, via marketplaces called exchanges where private insurers would sell health plans required to meet certain minimum standards" [15].
Funding
"'We are spending over $2 trillion a year on health care - almost 50 percent more per person than the next most costly nation,' he said during a nearly hour-long speech before the American Medical Association. 'For all this spending, more of our citizens are uninsured, the quality of our care is often lower and we aren't any healthier'" [16].
Other
"Deals are the lifeblood of legislation. Mary Landrieu of Louisiana got $100 million more for her state, Connecticut's Joe Lieberman stripped the bill of a government insurance plan and Ben Nelson won a slew of favors for Nebraska — all in exchange for their votes" [17].

E. Inter-Rater Reliability

The paragraphs were annotated by three coders trained in the classification criteria by the research team. Annotating was done independently by the two coders, and disagreements in coding were mediated by a research official. Non-parametric statistics were performed to calculate inter-rater reliability, which was 88% for subjectivity/objectivity coding, and 86.5% for dimension coding.

Table 2: Examples of Objective Paragraphs by Dimension

Stability and Security

"President Obama took his health care reform plan to the American people in a forum at the White House Wednesday night broadcast live on ABC. Obama took questions on a wide variety of topics, from how he plans to pay for the reform to whether people will be able to keep their current insurance plans and doctors" [18]

Quality and Affordability

"Massachusetts became the only state to mandate health insurance in 2006. It has passed legal muster and led to 97 percent of residents having some form of coverage, said Alan Sager, director of the Health Reform Program at Boston University's School of Public Health" [19].

Cost

"The House plan is projected to guarantee coverage for 96 percent of Americans at a cost of more than $1 trillion over the next 10 years, according to the nonpartisan Congressional Budget Office" [20].

Other

"Internet users looking for gift cards and other free merchandise are being steered to Web pages inviting them to send e-mails to Congress expressing their views on President Barack Obama's push to reshape the country's health system" [21].

Table 3: Annotation Guide for the Three Dimensions

Dimension	Dimension Themes
Stability and Security	
• "Ends discrimination against people with pre-existing conditions". • Prevents insurance companies from dropping coverage when people are sick and need it most. • Caps out-of-pocket expenses so people do not go broke when they get sick. • Eliminates extra charges for preventive care like mammograms, flu shots, and diabetes tests to improve health and save money. • Protects Medicare for seniors and eliminates the "donut-hole" gap in coverage for prescription drugs.	
Key Phrases: Pre-existing conditions, dropping coverage, out-of-pocket expenses, extra charges for preventative care, donut-hole, seniors, protecting Medicare	
Quality and Affordability	
• Creates a new insurance marketplace – the Exchange – that allows people without insurance and small businesses to compare plans and buy insurance at competitive prices. • Provides new tax credits to help people buy insurance and to help small businesses cover their employees. • Offers a public health insurance option to provide the uninsured who cannot find affordable coverage with a real choice. • Offers new, low-cost coverage through a national "high risk" pool to protect people with pre-existing conditions from financial ruin until the new Exchange is created.	
Key Phrases: Insurance marketplace, exchange, public option, provide uninsured with a real choice, high-risk pool, protect those with pre-existing conditions from financial ruin, low-cost coverage, coverage for all Americans, tax credits for businesses	
Funding	
• Will not add a dime to the deficit and is paid for upfront. • Creates an independent commission of doctors and medical experts to identify waste, fraud, and abuse in the health care system. • Orders immediate medical malpractice reform projects that could help doctors focus on putting their patients first, not on practicing defensive medicine. • Requires large employers to cover their employees and individuals who can afford it to buy insurance so everyone shares in the responsibility of reform.	
Key Phrases: Deficit, debt, spending, paid for upfront, waste, fraud, medical malpractice reform, large employers to cover employees, defensive medicine	
Other	
Paragraph not fitting any of the descriptions.	
Source: United States White House Office (www.whitehouse.gov)	

III. RESULTS

A. Overall Subjective and Objective Paragraphs in the Dataset

A comparison was conducted via Chi square test to compare more than proportions to determine any significant differences in the number of objective and subjective paragraphs among the writers' opinion. Overall, we found significantly more subjective paragraphs (59.7%, n = 1114) than objective paragraphs (16%, n = 281, p<0.0031). This trend was consistent in articles written before the ACA was passed (58.4% subjective, n = 594; 13.6% objective, n = 138), and after the ACA was passed (61.4% subjective, n = 520; 16.9% objective, n = 143; p<0.0031).

B. Overall dimension composition.

Comparing, in more detail, the distribution of subjective and objective comments on the three pre-defined dimensions, the analysis showed that overall, 16% (n = 299) of the paragraphs addressed Stability and Security; 59.7% being subjective and 16% objective. Paragraphs addressing Quality and Affordability accounted for 20.4% (n = 380) of the dataset, with 65.8% subjective and 9.6%, objective. The funding dimension was 17.6% (n = 328) of the sample, with 55.2% subjective and 13.9% objective. The rest of the sample (46%, n = 858) addressed a topic not related to the three dimensions. All of the dimensions had significantly more subjective than objective paragraphs (p<0.0001).

The Quality and Affordability dimension had significantly fewer objective paragraphs than the other dimensions (p<0.0001). Paragraphs that were under "Other" had more objective paragraphs than all of the pre-defined dimensions (p<0.0001).

C. Differences Between Articles Published Before and After the ACA was Passed

Chi square test was conducted to investigate the differences in dimension composition and the frequency of subjective and objective paragraphs between articles published before and after the ACA was passed. Table (5) provides the number of subjective and objective paragraphs for each dimension theme.

D. Dimension composition.

The Stability and Security dimension was mentioned more often in articles published after the ACA was passed (p<0.0001). The Quality and Affordability and Funding dimensions were mentioned more often in articles published before the ACA was passed (p<0.05).

Table 4: Sentiment Analysis of 105 Online News Articles Distributed on the Three Dimensions of Health Care Reform Plan

		Dimension				
	Sentiment Analysis	Stability/ Security	Quality/ Affordability	Funding	Other	Total
Disposition	Mix	84	117	80	270	551
		15.3%	21.2%	14.5%	49.0%	
		28.1%	30.8%	24.4%	31.5%	
	Neg	43	106	114	177	440
		9.8%	24.1%	25.9%	40.2%	
		14.4%	27.9%	34.8%	20.6%	
	Neut	74	54	67	240	435
		17.0%	12.4%	15.4%	55.2%	
		24.8%	14.2%	20.4%	28.0%	
	Pos	98	103	67	171	439
		22.3%	23.5%	15.3%	39.0%	
		32.8%	27.1%	20.4%	19.9%	
	Total	299	380	328	858	1865
	Chisq value 80.598; p <.0001					

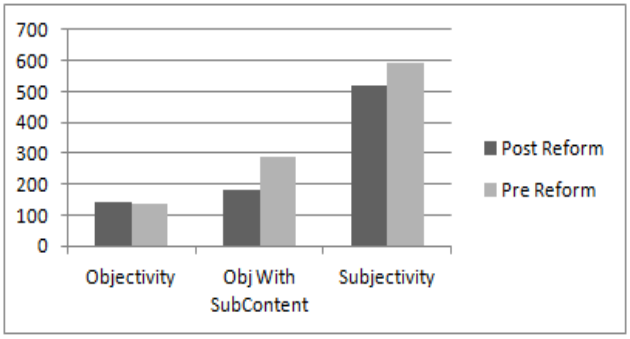


Figure 1. Count of subjective and objective paragraphs

Table5: Overall subjective and objective paragraphs by dimension

	Dimension				
	Stability/ Security	Quality/ Affordability	Funding	Other	Total
Objectivity	55	27	39	160	281
	19.6%	9.6%	13.9%	56.9%	
	18.4%	7.1%	11.9%	18.7%	
Obj With SubContent	79	103	108	180	470
	16.8%	21.9%	23.0%	38.3%	
	26.4%	27.1%	32.9%	21.0%	
Subjectivity	165	250	181	518	1114
	14.8%	22.4%	16.3%	46.5%	
	55.2%	65.8%	55.2%	60.4%	
Total	299	380	328	858	1865
Chisq value 46.810; p <.0001					

Table 6: Articles published before and After the ACA by dimension

Pre_Post	Dimension: Stability/ Security	Quality/ Afford	Funding	Other	Total
Post Reform	191	120	175	361	847
	22.6%	14.2%	20.7%	42.6%	
	63.9%	31.6%	53.4%	42.1%	
Pre Reform	108	260	153	497	1018
	10.6%	25.5%	15.0%	48.8%	
	36.1%	68.4%	46.7%	57.9%	
Total	299	380	328	858	1865
Chisq value 82.668; p <.0001					

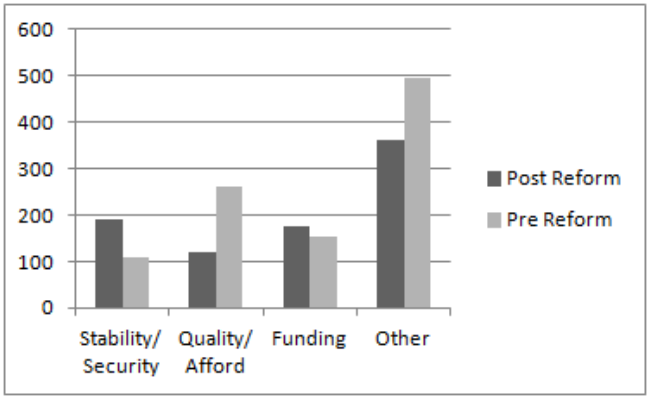


Figure 6: Articles published before and After the ACA by dimension

E. Differences Between Articles on Dimensions Before and After the ACA

Articles published before the ACA (Table 5 and Figure 5) was passed had significant differences among the plan dimensions proportions ($p<0.0001$).

IV. DISCUSSION

In this sample, subjective paragraphs greatly outnumbered objective paragraphs in online news articles. This finding suggests that online news media includes more subjective opinions than facts in their articles regarding health care reform. The above state of affairs may be due to the anonymity provided by online interaction – a situation that enables contributors to put forward claims even when such claims cannot be validated or can be validated with strategies or sources that are not based on provable or checked facts. These results imply that the public is receiving mostly subjective judgments when they are reading online news articles regarding health care reform. Thus, online news sources have the potential to expose the public to unreliable, politically motivated information that may be biased by the source of the news. This presence of unreliable information without concrete facts regarding health care reform may lead to mass confusion and anger (stemming from misinformation, misinterpretation, or biased interpretation) with the public because they are receiving conflicting news and coming into conflict with others who may have been exposed to an equally passionate, but opposite view.

The analysis of the content in each article suggests that more than half of the paragraphs addressed at least one of the dimensions – Stability and Security, Quality and Affordability, and Funding. This finding implies that members of the public are mostly receiving information that official government sources speak about. However, because the majority of these statements are subjective the public may not be receiving fact-based information about the official information from the White House, which may lead to misunderstandings and conflicts in the public regarding what the executive branch of the government is trying to say about health care reform.

When considering the differences between articles written before and after the ACA was passed, the Stability and Security dimension had a greater number of objective paragraphs post-reform. This may be due to the fact that the immediate effects of the ACA are to increase the stability and security of the insurance that individuals already receive and, therefore, the news may be more objective because it is a process that is actually being implemented. This finding suggests that online news articles may be more objective about health care reform if immediate action is being taken, but this finding should be interpreted with caution since the overall objective statements in post reform articles are low.

Conversely, the Quality and Affordability dimension has the lowest amount of objective statements, and there is no change between articles published before and after the ACA was passed. This may be because this dimension concerns how to implement reform so more Americans can be covered by health insurance. This dimension, therefore, has to do more with topics on which action may not have been taken, and on which there is considerable debate regarding what should be done. This finding implies that dimensions of health care reform that do not have legislation passed on them may have more objective statements in online news media because the public is debating about how that dimension should manifest itself, and the debates are being conducted with subjective opinions rather than objective facts. The lack of objective facts in debates regarding how health care reform should be implemented may be inhibiting the progress of health care reform as a whole.

Most of on-line news articles would be classified as "editorials" if they appeared in print news papers because they contained subjective classes.

This study has several limitations and these results should be interpreted with caution. We annotated the paragraphs rather than sentences in the articles, and did not look at the relationship between subjective and

objective statements in individual articles. Therefore, we were not able to infer the true intentionality of the authors writing the online news articles. In addition, based on our data, we cannot infer how the views of the American public regarding health care reform are influenced by reading online news media. Future research should investigate this relationship in order to illustrate exactly how the American public behaves when they are reading online news media, and how this may affect political outcome of future reform.

V. CONCLUSION

This study has demonstrated the significant importance of online news media in health communication. In particular, it has shown how powerful online news media can be in providing the American public with an avenue to voice facts and personal opinions on the most important health policy issue of this decade. In order for the American public to be educated on the facts about health care reform and have the ability to make their own informed decisions without a possible news bias, online news media outlets need to include more "objective" facts in their reporting. It is common to think of online news media as mostly for entertainment purposes, but since 50% (and growing) Americans get their news exclusively from the internet, online news publishers need to be accountable for the information they are making available to the public and provide more objective information.

REFERENCES

[1] L. Manchikanti and J. A. Hirsch, "Obama Health Care for All Americans: Practical Implications," *Pain Physician*, vol. 12, pp. 289-304, Mar-Apr 2009.

[2] L. R. Jacobs, "1994 all over again? Public opinion and health care," *New England Journal of Medicine*, vol. 358, pp. 1881-1883, May 1 2008.

[3] R. J. Blendon and J. M. Benson, "Understanding How Americans View Health Care Reform," *New England Journal of Medicine*, vol. 361, pp. E13-U18, Aug 27 2009.

[4] A. S. Chen and M. Weir, "The Long Shadow of the Past: Risk Pooling and the Political Development of Health Care Reform in the States," *Journal of Health Politics Policy and Law*, vol. 34, pp. 679-716, Oct 2009.

[5] A. Gelman, D. Lee, and Y. Ghitza, "Public Opinion on Health Care Reform," *Forum-a Journal of Applied Research in Contemporary Politics*, vol. 8, 2010 2010.

[6] S. E. Gollust, P. M. Lantz, and P. A. Ubel, "The Polarizing Effect of News Media Messages About the Social Determinants of Health," *American Journal of Public Health*, vol. 99, pp. 2160-2167, Dec 2009.

[7] T. W. House, "Health Reform in Action," 2010.

[8] J. Wiebe, R. Bruce, and T. O'Hara, "Development and use of a gold standard data set for subjectivity classifications," in *In Proc. 37th Annual Meeting of the Assoc. for Computational Linguistics (ACL-99)*, 1999, pp. 246-253.

[9] M. Marcus, B. Santorini, and M. Marcinkiewicz, "Building a large annotated corpus of english: The penntreebank," *Computational Linguistics*, vol. 19, pp. 313-330, 1993.

[10] R. Bruce and J. Wiebe, "Recognizing subjectivity. A case study of manual tagging," *Natural Language Engineering*, vol. 5, 1999.

[11] J. Wiebe, "Learning subjective adjectives from corpora," in *In Proc.17th National Conference on Artificial Intelligence (AAAI-2000)*, 2000, pp. 735-741.

[12] E. Riloff, J. Wiebe, and W. Phillips, "Exploiting subjectivity classification to improve information extraction," in *In Proc. 20th National Conference on Artificial Intelligence (AAAI-05)*, 2005, pp. 1106-1111.

[13] J. Hurst, "Effective ways to realise policy reforms in health systems," OECD iLibrary2010.

[14] M. Feldstein, "Obama's plan isn't the answer.," in *The Washington Post*, ed, 2009.

[15] E. Werner, "Dems, White House predict success on health care," in *The Associated Press*, ed, 2009.

[16] B. Japsen, J. McCormick, and N. N. Lavey, "President Barack Obama gives doctors group his health-care Rx; U.S. system is a 'tricking time bomb' threatening prosperity, president tells AMA," in *Chicago Tribune*, ed, 2009.

[17] R. Zaldivar-Alonso, "Abortion remains a key obstacle to final passage of health care bill," in *The Associated Press*, ed, 2009.

[18] R. Klein, B. Hovell, and B. Z. Wolf, "Fact check: the truth behind Obama's health care plan," in *ABC news*, ed, 2009.

[19] K. Wyatt, "Car insurance scofflaws raise health mandate doubt," in *The Associated Press*, ed, 2009.

[20] B. Keilar, "Differences remain over what health care bill will look like," in *CNN news*, ed, 2010.

[21] T. A. Press, "Internet users lured to oppose health bills," ed, 2009.

Automatic Construction of Similarity Matrix for Semantic Numerical Operations on Strings

Taeho Jo
School of Computer and Information Engineering
Inha University
Incheon, South Korea
tjo018@inha.ac.kr

***Abstract*—This research is concerned with the simulation of the numeric semantic operations on strings based on the similarity matrix constructed from the collection of news articles. The motivation of this research is invention of string vector based machine learning approaches to text mining tasks. In this research, we define and simulate the three operations: 'semantic similarity', 'semantic similarity average', and 'semantic similarity variance'. This research is expected to become the basis from which semantic analysis tools or systems of words, texts or corpus, are developed as its benefit. We present the simulations of carrying out operations on strings in the real corpus: 20NewsPageGroups.**

***Keywords*-semantic operation, Similarity Sematic Average, Similarity Semantic Variance**

I. Introduction

The semantic operations refer to the operations based on meanings of strings rather than their lexical properties. The words in textual data are given as operands of the operations which were proposed in this research. As the basis of performing the operations, we use the similarity matrix which consists of semantic similarities indicating how much corresponding words are similar as each other. In this research, we define the three operations: 'semantic similarity', 'semantic similarity average', and 'semantic similarity variance'. The proposed operations should be distinguished from the lexical operations on strings based on their spellings

Previously, we attempted to replace numerical vectors by string vectors in representing texts. The reason of the replacement is the three problems: huge dimensionality, sparse distribution, and poor transparency; they are described in detail in the literatures [1][2][3][4][5]. The replacement leads to the successful performance in text categorization and clustering. However, in order to use the string vectors more naturally and freely, we need more systematic mathematical analysis and definitions on strings. The previous research concerned with encoding of texts into string vectors will be mentioned in section 2.

In this research, we define the three semantic operations on strings. The semantic similarity between two words indicating how much two words are similar as each other, is included as the basic operation. The SSA (Semantic Similarity Average) is proposed as the average over similarities of all possible pairs of words. From the SSA, we derive SSV (Semantic Similarity Variance) as the variance over the similarities. In this research, we call the defined operations numerical semantic operations, since numerical values are generated from the operations as their outputs.

We expect the three benefits from this research. For first, the semantic operations are potentially used for developing string vector based approaches to tasks of text mining and information retrieval. For second, this research may provide the basis for developing automatic semantic analysis tool for words and texts. For third, the possibility of developing even digital computers only for text processing is available potentially. In order to take the benefits, we need to define more semantic operations and characterize them mathematically.

This article is composed of the five sections. In section 2, we explore the previous research relevant to this research. In section 3, we describe the proposed semantic operations formally and characterize them mathematically. In section 4, the operations are simulated on the real corpus. In section 5, as the conclusion, we mention the significances and the remaining tasks of this research.

II. Previous Works

This section is concerned with the exploration for the previous works relevant to this research. In 2000, Jo invented a new neural network, proposing encoding documents into string vectors; it provides the motivation for doing this research [1]. The semantic relations between words are considered for doing information retrieval tasks such as ranking and term weighting. Even for doing other tasks, the semantic relations are also considered. Therefore, in this section, we will explore previous works in terms of string vector encoding and tasks involving the semantic relations between words.

This research is initiated from encoding documents into string vectors, instead of numerical vectors, for doing text mining tasks. Encoding documents so was initiated by Jo in 2000, inventing the new neural network, called NTC (Neural Text Categorizer), as a practical approach to text categorization [1]. Subsequently, in 2005, Jo and Japcowicz invented the unsupervised string vector based neural network

which was called NTSO (Neural Text Self Organizer) [6]. In 2009, Jo modified the KNN and SVM into its string vector based versions where the similarity measure between string vectors was based on the semantic relations between words [7]. However, in order to use string vectors more freely, we need to define more semantic operations on strings and characterize them mathematically.

The semantic relations between words are considered especially in information retrieval tasks. In 2005, Shenkel et al implemented the search engine which was called XXL, for searching for XML documents, using the semantic similarity between words, based on ontology and an index structure [8]. In 2005, Possas et al proposed a term weighting scheme which was called 'set based model', considering the semantic relation between term [9]. In 2008, Vechtomova and Karamuftuoglu used the semantic relation between a query and terms for ranking retrieved documents [10]. The previous works show usefulness of the semantic relation between words in the domain of information retrieval.

The semantic relation between words may be considered in other tasks as well as the information retrieval tasks. In 1994, Kiyoki et al defines metadata of image as words for representing their semantic relations for the image retrieval [11]. In 2004, Makkonen et al defines semantic relations among words using ontology for doing the topic tracking and detection [12]. In 2007, Na et al used the semantic relation between a query and terms for adjusting clustering results [13]. The previous works show that the semantic relation may be considered in various tasks.

This research is intended to define various semantic relations between words, assuming that each string has its own meaning. In the previous works, the semantic relations have been considered not mathematically but informally or implicitly. In other words, the mathematical foundations are not founded, yet; the computation of the semantic similarity has depended on very heuristic computations. Even if the modification and creation of string vector based approaches in favor of text categorization and clustering was successful, it was limited to process string vectors because of no more systematic mathematical foundations. Therefore, the goal of this research is to define more semantic operations on strings and characterize them algebraically, in order to overcome the limitation.

III. Numerical Semantic Operations

This section describes the semantic operations in detail and consists of the four sections. In section 3.1, we describe the similarity matrix as the basis of carrying out the semantic operations. In section 3.2, we mention the two opposite operations: semantic similarity and semantic distance. In section 3.3, we define the SSA formally and characterize it mathematically. Section 3.4 covers the SSV like the SSA.

A. Similarity Matrix

Before entering the semantic operations on strings, we will describe the similarity matrix in this section. The similarity matrix is used as the basis for performing the semantic operations on strings. In the similarity matrix, each of its rows and columns corresponds to a string. The matrix has the two properties: its elements are symmetry and its diagonal elements are 1s. The similarity matrix defines the semantic similarity of each of all possible pairs of strings, and it assumes that the matrix is always given before doing the operations on strings.

The similarity matrix refers to the square matrix which defines the semantic similarity of each of all possible pairs of strings, and it is denoted as follows:

$$\begin{pmatrix} s_{11} & s_{12} & \cdots & s_{1N} \\ s_{21} & s_{22} & \cdots & s_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ s_{N1} & s_{N2} & \cdots & s_{NN} \end{pmatrix}$$

The similarity matrix is given as the N by N matrix, and N indicates the total number of strings. Each of the columns and the rows corresponds to its unique string; both the ith row and the ith column correspond to the identical string. The element of the similarity matrix, s_{ij} indicates the semantic similarity between the string corresponding to the ith column and that corresponding to the jth row. Following the two properties, the similarity matrix may be built manually or automatically.

The first property of the similarity matrix is that its elements are symmetry to each other. In other words, the rule $s_{ij} = s_{ji}$ applies to all elements in the similarity matrix. We already mentioned that the string corresponding the ith column is identical to that corresponding to the ith row. The two strings which correspond to the ith column and the jth column is same to those which correspond to the vice versa. The commutative raw is applicable to the semantic similarity between two strings.

The second property of the similarity matrix is that its diagonal elements are always given 1.0. In other words, s_{ii} is given as 1.0 as the maximum similarity. Every element in the similarity matrix is given as a normalized value between 0 and 1. The value, 1.0, signifies the maximum similarity between two strings. In the context of this research, it is assumed that the two identical strings have their maximal similarity.

The similarity matrix may be constructed, manually or automatically. A finite set of strings and the size of the similarity matrix are decided in advance. The 1.0 values are absolutely assigned to the diagonal elements of the similarity matrix. Keeping its symmetry property, normalized values between 0 and 1 are assigned to the off-diagonal elements. In other literatures, the process of building the similarity

matrix from a corpus is mentioned; refer to the literatures for the detail description of the automatic construction.

B. Semantic Similarity and Distance

This subsection is concerned with the two base semantic operations on strings. One operation covered in this section is for evaluating how much two strings are similar based on their meanings. The other is for doing how much they are different from each other with respect to their meanings. The commutative law is applicable to both operations; the result is identical to the different order of the input strings. Therefore, in this subsection, we will describe the both operations with respect to their definition and properties.

The first base operation is for evaluating a semantic similarity between two strings. We already described the similarity matrix in section 1, as the basis of these operations. It is possible to construct automatically the similarity matrix from a corpus, but the detail process is not covered in this article. As shown in figure 1 The semantic similarity is carried out by retrieving directly the corresponding element from the similarity matrix as follows:

$$sim(str_i, str_j) = s_{ij}$$

This operation becomes the fundamental one for deriving more advanced operations, later.

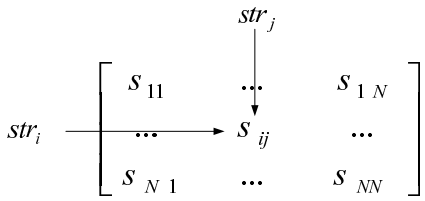


Figure 1. The Process of Retrieving the Semantic Similarity from the Similarity Matrix

The second operation is the semantic distance which is opposed to the previous operation. Like the semantic similarity, this operation generates a normalized value between 0 and 1 as the output. The semantic distance between two strings is computed by subtracting the semantic similarity from 1.0 as follows:

$$dis(str_i, str_j) = 1.0 - sim(str_i, str_j) = 1.0 - s_{ij}$$

The value generated from the semantic distance is the 1.0's complement of the semantic similarity. We may build the semantic distance matrix by subtracting each element from 1.0 as follows:

$$\begin{pmatrix} 1.0 - s_{11} & 1.0 - s_{12} & \dots & 1.0 - s_{1N} \\ 1.0 - s_{21} & 1.0 - s_{22} & \dots & 1.0 - s_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ 1.0 - s_{N1} & 1.0 - s_{N2} & \dots & 1.0 - s_{NN} \end{pmatrix}$$

Both operations are characterized as the fact that the commutative law is applicable. In the case of the semantic similarity, the commutative law applies because the similarity matrix is symmetry, as follows:

$$sim(str_i, str_j) = s_{ij} = s_{ji} = sim(str_j, str_i)$$

The commutative law also applies because the same value is subtracted from 1.0 as follows:

$$dis(str_i, str_j) = 1.0 - s_{ij} = 1.0 - s_{ji} = dis(str_j, str_i)$$

The similarity distance matrix becomes symmetry, but its diagonal elements are 0 values instead of 1.0 values. If the similarity distance matrix is given, the semantic distance is carried out by retrieving the corresponding element from the matrix.

C. Semantic Similarity Mean

This subsection is concerned with the first n-ary semantic operation on strings. The n-ary semantic operation refers to the class of semantic operations which takes an arbitrary number of strings as the input. In this operation, all possible pairs of strings are generated and the semantic similarity to each pair is computed. The semantic similarity mean of the strings is computed by averaging the similarities of the all possible pairs. In this subsection, we will describe the operation with respect to the definition, the properties, the procedure, and the utility.

This operation is denoted as follows:

$$avgsim(str_1, str_2, \dots, str_n) = \frac{2}{n(n-1)} \sum_{i<j} sim(str_i, str_j)$$

When n strings are given as the input, we generate $n(n-1)/2$ pairs of strings as all possible ones. For each pair, we may compute the similarity by retrieving it from the similarity matrix as shown in figure 1. We obtain the average semantic similarity by summing the similarities of all pairs and dividing the sum by the number of all possible pairs, $n(n-1)/2$. The average semantic similarity signifies the semantic cohesion of the group of strings.

The properties of this operation are as follows:

- If all strings are identical, the average semantic similarity is given as 1.0 values, since the diagonal elements of the similarity matrix are given 1.0.
- $\frac{2}{n(n-1)} \sum_{i<j} sim(str_i, str_j) = \frac{2}{n(n-1)} \sum_{i>j} sim(str_i, str_j)$, since the similarity matrix is symmetry one.
- If all pairs of the strings are complementary (lowest similarity), the average semantic similarity becomes the minimum.
- The average semantic similarity is always given as a normalized value, since the similarities of all possible pairs are given as normalized values.

This operation takes an arbitrary number of strings as the input. Among the strings, all possible pairs are generated; if the number of strings is n, $n(n-1)/2$ pairs are generated.

For each pair, its similarity is retrieved from the given similarity matrix. The average semantic similarity is obtained by summing the similarities of the all possible pairs and dividing the sum by the number of pairs. Therefore, the averaged semantic similarity which is given as a normalized value is the output of the operation.

Figure 2 illustrate the two different groups of strings. The left group in figure 2 contains the strings within the domain of computer science. The right group in figure 2 contains the strings spanning over various domains. Intuitionally, the left group of strings has the higher average semantic similarity than the right group. Through the example illustrated in figure 2, this operation may be used for estimating the cohesion of groups of strings.

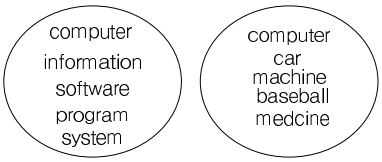


Figure 2. Two Groups of Words: One in a specific domain and the other in various domains

D. Semantic Similarity Variance

This subsection is concerned with the second n-ary semantic operation on strings. Under an identical average semantic similarity, there exist different distributions of similarities of pairs of strings. The pairs of strings may concentrate on the average semantic similarity, or they may disperse from it. We need the measure how much the similarities of the pairs concentrate on the average semantic similarity. In this subsection, we describe the operation in detail with respect to its definition, properties, and procedure.

The operation, called semantic similarity variance, is denoted as follows:

$$var(str_1, \ldots, str_n) = \frac{2}{n(n-1)} \sum_{i<j} (sim(str_i, str_j) - avgsim(str_1, \ldots, str_n))^2$$

If n strings is given as the input, the number of all possible pairs becomes $n(n-1)/2$. Before performing this operation, the average semantic similarity should be computed by the operation which was mentioned in section III-C. This operation focuses on the individual square of difference between a similarity of each pair and the average semantic similarity. This operation corresponds to the variance in the context of statistics.

The properties of this operation are as follows:

- $\frac{2}{n(n-1)} \sum_{i<j} (sim(str_i, str_j) - avgsim(str_1, \ldots, str_n))^2 = \frac{2}{n(n-1)} \sum_{i>j} (sim(str_i, str_j) - avgsim(str_1, \ldots, str_n))^2$
- $sd(str_1, \ldots, str_n) = \sqrt{var(str_1, \ldots, str_n)}$

$sd(str_1, \ldots, str_n)$ is called the semantic similarity standard deviation.

In this operation, an arbitrary number of strings is given as the input. Using the operation which was mentioned in section III-C, the semantic similarity average is computed. For each pair, the difference square between its similarity and the average semantic similarity is computed. The difference squares are averaged into the semantic similarity variance. The square root of the semantic similarity variance becomes the semantic similarity standard deviation. Whether it is the variance or standard deviation, the value is always given as a normalized value.

The operation may be used for judging whether words are distributed, randomly or not. Let's consider the two groups of words with their identical semantic similarity. One group whose semantic similarities are concentrated on the average semantic similarity has very small the semantic similarity variance. However, the other whose semantic similarities are dispersed very much has the larger semantic similarity variance. In this case, the latter group is judged as the random distribution of words.

IV. Simulations

This section is concerned with a set of simulations of carrying out the semantic operations based on another similarity matrix. The similarity matrix is constructed from another corpus called '20NewsGroups' in this set of simulation. Like the previous set of simulations, the similarities among words are computed based on the texts where words collocate with each other. The set of simulations are carried out, following the process mentioned in section **??**. In this section, we present and discuss the set of simulation.

In this set of simulation, we use 20NewsGroups which is a collection of news articles as the source of building the similarity matrix. The collection was obtained by downloading from the web site, http://kdd.ics.uci.edu /databases/20newsgroups/20newsgroups.html. In the collection, 20 categories are predefined and approximately 20,000 news articles are available. The collection was intended for researchers on text mining to evaluate approaches to text categorization. The set of simulation was executed, following the steps which were mentioned in section **??**.

This set of simulations is carried out with three steps: indexing the corpus, constructing the similarity matrix, and carrying out the semantic operations on strings. The corpus, which is the collection of texts, is indexed into a list of words and their frequencies as shown in figure 1. We selected 100 words randomly and built the 100 X 100 similarity matrix by computing semantic similarities among words based on the number of texts where the words collocates with each other. We made 16 lists each of which consists of five words by selecting words randomly among the selected 100 words and applied the semantic similarity average and variance to

each list. We generated values of the semantic similarity averages and variances as results of this set of simulations.

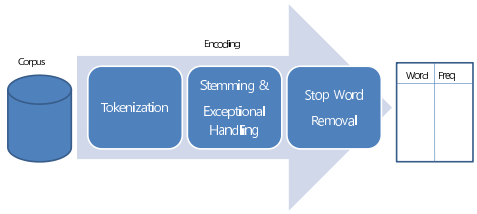


Figure 3. The Process of Indexing Corpus

In figure 4, we illustrate the results from simulating the operations based on the similarity matrix constructed from 20NewsGroups in figure 4. We present the 16 lists each of which consists of five words. The values of SSA and SSV are indicated by the black and white bar, respectively. The six lists have SSA which is greater than or equal to 0.1, and the two lists have SSV which is so. The others have values of SSA and SSV less than 0.1.

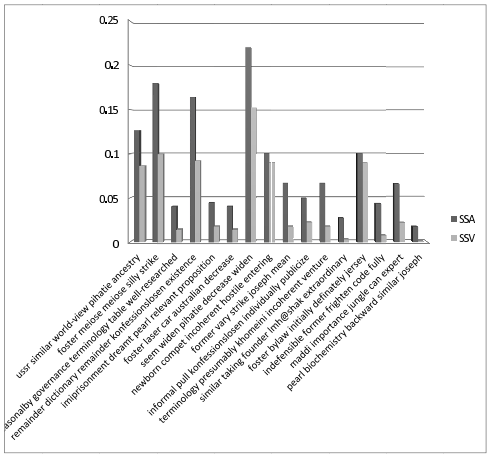


Figure 4. The Simulation Results from the SSM and SSV from the Corpus: 20NewsGroups

Let's discuss the simulation results which are illustrated in figure 4. The list which contains seem, widen, phihates, decrease, and widen, has the SSA which is higher than 0.2 and the SSV which is higher than 0.15. It indicates that the majority of words have their strong semantic relationships, together with the minority of words with their weak semantic relationships. The list which contains foster, bylaw, initially, definitely, and jersey, has its high SSV compared with its SSA. It is characterized as the coexistence of weak and strong semantic relationships among words.

V. Conclusion

Let's consider the significances of this research. From this research, we obtain the chance to measure semantic relations among words by the two simple operations: semantic similarity and semantic distance. We are able to observe the semantic cohesion of words through the operation called SSA. It gets possible to observe the distributions over semantic similarities of words, through the operation called SSV. This research provides potentially the way of developing semantic analyzer of textual data.

In spite of the above significances, let's consider remaining tasks for proceed further research. We need to make more simulations of carrying out the operations in other domains. More semantic operations will be defined and characterized mathematically. When the complexity of performing the semantic operation is high, it is necessary to reduce the complexity by developing their approximating algorithms. The operations will be applied to text processing tasks in information retrieval systems and text mining systems.

References

[1] T. Jo, "NeuroTextCategorizer: A New Model of Neural Network for Text Categorization", pp280-285, The Proceedings of ICONIP 2000, 2000.

[2] T. Jo, "The Implementation of Dynamic Document Organization using Text Categorization and Text Clustering", PhD Dissertation of University of Ottawa, 2006.

[3] T. Jo and D. Cho, "Index Based Approach for Text Categorization", pp127-132, International Journal of Mathematics and Computers in Simulation, Vol 2, No 1, 2008.

[4] T. Jo, "Topic Spotting to News Articles in 20NewsGroups with NTC", pp50-56, Lecture Notes in Information Technology, Vol 7, 2011.

[5] T. Jo, "Definition of String Vector based Operations for Training NTSO using Inverted Index", pp57-63, Lecture Notes in Information Technology, Vol 7, 2011.

[6] T. Jo and N. Japkowicz, "Text Clustering using NTSO", pp558-563, The Proceedings of IJCNN, 2005.

[7] T. Jo, "Modification of Classification Algorithm in Favor of Text Categorization", pp13-23, International Journal of Computer Science and Software Technology, Vol 2, No 1, 2009.

[8] R. Schenkel, A. Theobald, and G.Weikum, "Semantic Similarity Search on Semistructured Data with the XXL Search Engine", pp521-545, Information Retrieval, Vol 8, No 4, 2005.

[9] B. Possas, N. Ziviani, W. Meira, Jr., and B. Ribeiro-Neto, "Set-based vector model: An ecient approach for correlation-based ranking", pp397-429, ACM Transactions on Information Systems, Vol. 23, No. 4, 2005.

[10] O. Vechtomova and M. Karamuftuoglu, "Lexical cohesion and term proximity in document ranking", pp1485-1502, Information Processing & Management, Vol 44, No 4, 2008.

[11] Y. Kiyoki, T. Kitagawa and T. Hayama, "A Metadatabase System for Semantic Image Search by a Mathematical Model of Meaning", pp34- 41, ACM SIGMOD Record, Vol 23, No 4, 1994.

[12] J. Makkonen, H. Ahonen-Myka, and M. Salmenkivi, “Simple Semantics in Topic Detection and Tracking”, pp347-368, Information Retrieval, Vol 7, No 3-4, 2004.

[13] S. Na, I. Kang and J. Lee, “Adaptive document clustering based on query-based similarity”, pp887-901, Information Processing & Manage- ment, Vol 43, No 4, 2007.

Mining Social Data with UCL's SocialSTORM Platform

R. Wood, I. Zheludev, and P. Treleaven
UK PhD Centre in Financial Computing, University College London, London, United Kingdom

Abstract - *SocialSTORM is a cloud-based 'central-hub' which facilitates the acquisition, storage and analysis of live data from social media feeds. Developed at University College London, the platform manages data from Twitter, Facebook, RSS sources and blogs, and is currently being extended to other feeds. This is for the purpose of harvesting information which ceases being publically-available shortly after creation. SocialSTORM includes facilities to run simulation models on the data allowing for the identification of changing trends, global sentiments and story propagation. Both historical and live data streams can be monitored. We are specifically interested in using these data for trading applications, although it has applicability to security monitoring, brand awareness and macroeconomic variable monitoring.*

Keywords: Social Science, Web Mining, Mining text and semi-structured data, Mining large scale data, Data mining software

1 Introduction

Social media is becoming a rich new area of research for scientific, sociological and commercial purposes. Acknowledging its huge value, it is likely that companies such as Google, Facebook and Twitter will increasingly restrict access to their social media data. With the rise of Computational Social Science, arguably what is required to support academic research is a public-domain social data scraping and analytics environment and high-performance computing facility.

To capitalize on the wealth of data currently available across the web, for the purposes of academic research, University College London (UCL) has built, and continues to develop, a comprehensive social media data engine that supports scraping and analysis of a wide range of social media data. This paper describes this social media Streaming, Online Repository and Analytics Manager (SocialSTORM) platform[1]. It can be seen as a multi-source customizable research-orientated counterpart to commercial social aggregation systems such as Datasift[2] and GNIP[3]. To our knowledge there are no similar social data acquisition and monitoring platforms tailored specifically to the research community. The closest equivalent we have found is Wandora[4], an open source information extraction, aggregation and data management system designed to run on a local machine. It is not directly suited to large-scale data mining, but via the creation of proprietary Java code, it can be used to monitor Twitter data.

2 Mining large scale data

2.1 Web mining for social data

Currently, large quantities of public data from sources such as Twitter and Facebook can be acquired free of charge. Data are typically accessed by querying an Application Programming Interface (API) for each of these respective social media providers; and this may be used to tailor results according to a desired dataset via proprietary code. Twitter for example, allows developers to track up to 400 specified keywords for which to filter publicly available updates before streaming to the developer in near real-time. It is also possible to filter Twitter data by user ID or location, achievable with a simple HTTP POST request. Obtaining a 'random sample' of data from Twitter is even easier; the following HTTP GET request returns a live stream roughly 1% of all public status updates as a JSON array:

https://stream.twitter.com/1/statuses/sample.json

Furthermore, elevated access to a random sample of approximately 10% of all global Tweets is straightforward to obtain for academic research purposes. Once these Twitter data have been published and streamed through its API, the data ceases to be accessible from Twitter. This highlights the need for continuous communication with Twitter, and suitable technologies for storage of the data to allow aggregation of a substantial dataset over time (discussed later).

Facebook also offers an API through which publicly available data are accessible; though not in real-time. It is also less common for Facebook users to make their updates publicly visible – this is in contrast to Twitter's policy where Tweets are automatically in the public domain by default (with an opt-out option). However, given Facebook's user-base of c. 800 million people, it is still reasonable to expect large volumes of data to be available for retrieval and

[1] www.socialstorm.eu
[2] www.datasift.com
[3] www.gnip.com
[4] www.wandora.org

analysis. Facebook's Graph API can be used to search for status updates containing particular keywords specified by the developer; these results go back as far as 70 days from the request date. The following HTTP GET request can be used to access these data, returning a JSON array of data relating to all public posts containing the term 'Apple' used here as an example:

https://graph.facebook.com/search?q=Apple&type=post

Through the same API it is also possible to retrieve public data relating to Facebook *Pages*, *Events*, *Users*, *Groups*, *Places* or *Checkins* by modifying the 'type' parameter in the above URL accordingly.

A 'random sample' of all public updates from Facebook can also be harvested, by using the query search function to look for a collection of a language's most commonly used words such as: 'to', 'be', 'and', 'of' if working in English.

2.2 Mining text and semi-structured data

Before analyzing social media data from Facebook and Twitter, one must extract the relevant data fields from their raw JSON format. Fig. 1 provides an example of the information fields returned for each Tweet retrieved via the filtering method of Twitter's Streaming API (data has been removed to protect privacy).

```
{
      "text":"",
      "entities":{},
      "contributors": ,
      "place": ,
      "id_str":" ",
      "coordinates": ,
      "source":" ",
      "retweet_count": ,
      "in_reply_to_user_id": ,
      "in_reply_to_status_id": ,
      "favorited": ,
      "geo": ,
      "in_reply_to_screen_name":" ",
      "truncated": ,
      "in_reply_to_status_id_str":" ",
      "user":{},
      "retweeted": ,
      "id": ,
      "in_reply_to_user_id_str":" ",
      "created_at":" "
}
```

Fig. 1. Example response from Twitter's Streaming API

Twitter offers many forms of metadata which may also provide a source for analysis, as well as the Tweet ("`text`") itself. Examples include location tags and the number of times the message was "retweeted" (shared by other users, thus increasing the audience). These may consist of integers, strings, or a combination of both. In order to extract these data, one needs to parse the raw JSON structure and store each desired string or integer as a separate variable. Conveniently, one may use the very same method to structure data retrieved from Facebook (which is also in JSON by default). The data may then, for example, be stored within separate columns of a database, or as a text file with a specified delimiter. It is inadvisable to store this kind of data as a text file in CSV (comma-separated variable) format since status updates themselves frequently contain one or more commas which can make subsequent analysis tricky.

Text data may be analyzed in a number of ways, using techniques from Natural Language Processing and Information Retrieval. An obvious direction to take when analyzing text-based social media data is to conduct sentiment analysis in order to quantify message strings, to enable mathematical models to be employed for analysis.

3 SocialSTORM

3.1 Overview

University College London, assisted by Microsoft, has built a cloud-based computational finance environment (ATRADE[5]) that supports real and virtual trading; with terabytes of financial data to support research into algorithmic trading and risk. Given the rise of interest in using social data (e.g. Twitter updates) for trading and risk management, UCL has now built SocialSTORM, a complementary social media engine that supports scraping and analysis of a wide range of social media data.

As discussed, SocialSTORM is a cloud-based platform which facilitates the acquisition of text-based data from online sources such as Twitter, Facebook, respected blogs, RSS media and 'official' news; a 'central-hub' for social media analytics. The system includes facilities to upload and run Java-coded simulation models to analyze the aggregated data; which may comprise UCL's social data and/or users' own proprietary data. There is also connectivity to the ATRADE platform which provides further quantitative finance and economic data.

The platform consists of infrastructure and tools to facilitate data acquisition, database connectivity, and various levels of access and administration along with data repositories for long and short-term data storage. The platform is able to operate in two simulation modes: an 'historical' mode which utilizes data already stored at the time of running the desired simulation (ideal for data-mining and back-testing), and a 'live' mode which operates on a near real-time stream of data which is continually monitored from the sources throughout the simulation (ideal for analyzing financial markets and developing algorithmic trading strategies).

[5] http://vtp.cs.ucl.ac.uk/atrade

In short, SocialSTORM allows for the execution of user-defined simulation models for the analysis of historical and real-time data feeds that provide a plentiful supply of public opinions derived from online-community data.

3.2 Infrastructure architecture

The SocialSTORM platform resides in a distributed computing environment currently consisting of 9 nodes each with the following specification: 15,000rpm 600GB hard drive, 32GB RAM and one 3.2GHz quad-core Intel Xeon e3-1200 processor. The nodes are interlinked by 10GbE (10 Gigabit Ethernet) connections and the entire system is backed-up daily onto tape storage for up to 3 months. SocialSTORM's current storage capacity is 5.4TB with 288GB of available RAM. SocialSTORM is fully scalable – additional nodes can be added to increase system storage and performance on an as-needed basis.

This particular hardware setup has been chosen for the purposes of migrating SocialSTORM to Apache Hadoop, a software library and framework that allows for the distributed processing of large data sets[6]; which is something we are currently working towards.

3.3 System architecture

SocialSTORM inherits its architectural design from UCL's ATRADE system, which allows easy integration between the two systems. The following is an outline of the key components of the SocialSTORM system.

Connectivity Engines – Various connectivity modules communicate with the external data sources, including Twitter & Facebook's APIs, financial blogs and various RSS news feeds; and are being continually expanded to incorporate new social media sources. Data are fed into SocialSTORM in real-time and include a random sample of all public updates from Twitter, as well as filtered data streams selected from a rich dictionary of stock symbols, currencies and other economic keywords; providing gigabytes of text-based data every day.

Messaging Bus – This serves as the internal communication layer which accepts the incoming data streams (messages) from the various connectivity engines, parses these (from either JSON or XML format) and writes the various data to the appropriate tables of the main database.

Data Warehouse – This is home to terabytes of text-based entries which are accompanied by various types of metadata to expand the potential avenues of research. Entries are organized by source and accurately time-stamped with the time of publication, as well as being tagged with topics for easy retrieval by simulation models.

Simulation Manager – This terminal provides the external API for clients to interact with the data for the purposes of analysis, including a web-based GUI via which users can select various filters to apply to the datasets before uploading a Java-coded simulation model to perform the desired analysis on the data. The Simulation Manager facilitates all client-access to the data warehouse, and also allows users to upload their own datasets for aggregation with UCL's social data for a particular simulation. There is also the option to switch between historical mode (which analyses data existing at the time the simulation is started) or live mode (which 'listens' to incoming data streams and performs analysis in real-time).

In summary, the aims of SocialSTORM include acquisition and access to terabytes of social data from a variety of sources, as well as a cloud-based simulation environment for historical data-mining and real-time monitoring of global news and opinions taken from the world's most popular social networking sites. There is also connectivity to UCL's ATRADE algorithmic trading system and support for aggregation of clients' proprietary data. UCL's cloud-based platform removes the need to transfer large amounts of data across servers and also eliminates dependency on the processing power of clients' local machines; leading to increased performance in working with 'Big' datasets.

3.4 Data storage

SocialSTORM queries and monitors social media APIs in real-time, reading updates as they are streamed and writing these directly to its database. The latency between a message being published to Twitter (as an example) and subsequently being stored in our database is less than 1 second; even when using batch inserts to increase efficiency. Typically, the system writes c. 4,000 entries to the database every second.

From Twitter the current system retrieves c. 20 million messages per day as a 'random sample' of all public updates, plus c. 1-2 million messages daily containing hundreds of specific financial and economic keywords selected by the platform's development team. From Facebook, a proprietary method of retrieving a random sample of all public updates is used which returns c. 2 million updates per day. This is supplemented by searching for updates containing the same keywords used to filter updates from Twitter; giving over 500,000 additional daily updates from Facebook. The SocialSTORM team has selected 15 finance-related blogs to monitor, as well as a number of official news services which, together contribute over 1,000 daily entries to the database.

The current data sources result in approximately 5GB of data per day, which is likely to continue to increase over time barring any restriction to public data by Social Media companies; UCL has the facilities to cope with an increased dataflow. The current SocialSTORM servers allow storage of

[6] http://hadoop.apache.org/

multiple terabytes of data; but may be scaled-up to petabytes if required.

3.5 Simulations

User-privacy is taken very seriously by the platform's development team. Although the data retrieved from the web is in the public domain, it remains property of the data provider and is therefore not redistributable in accordance with Content License Agreements. To enable analysis of social media data by third parties, SocialSTORM includes a black-box research environment, accessible via a graphical web interface as shown in Fig. 2. Here, subscribed users may upload their own java-coded simulation models which will analyze the data stored by SocialSTORM and return post-analytical results to the caller according to the model used.

Models to be uploaded to SocialSTORM are **.jar** files, which also include any packages on which the code is dependent. The simulation environment then looks for a particular method, similar to **Main()**, which defines the appropriate parameters to interface with SocialSTORM's API. Instructions on how to ensure that models are compliant with the platform are detailed in the SocialSTORM user manual.

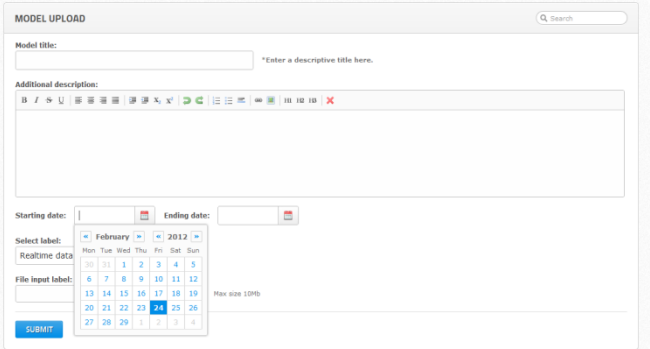


Fig. 2. Model-upload form for SocialSTORM simulations

Before running a model the user can opt to perform the analysis in 'live' mode, which connects directly to the platform's real-time messaging system to stream live updates to the model, or 'historical' mode which retrieves data already stored in the database. The user may choose to pause or stop a simulation at any time, and a 'live' simulation is complete when a certain breakpoint in the code is reached or until the user manually stops the simulation. Once a simulation is complete, users can plot results in various ways using the SocialSTORM GUI (an example of which is shown in Fig. 3), export results to Microsoft Excel, or use an output API to retrieve the results programmatically for further analysis. Data exported to Microsoft Excel can be linked to constantly update in a spreadsheet's cells.

4 Applications

The applications of social media data in academia and commerce are growing rapidly. As Computational Social Science expands, platforms such as SocialSTORM should provide useful research tools. To demonstrate the wider appeal of aggregated social media data, we present the following specific examples.

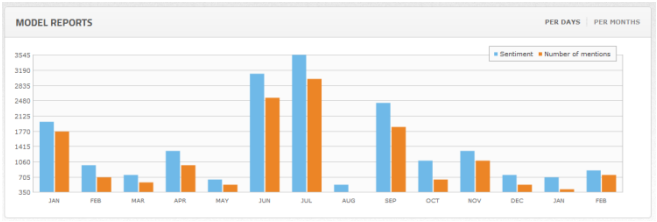


Fig. 3. Example of a bar chart plot of simulation results in SocialSTORM's web interface

4.1 Social science

The analysis of narratives of professional journalistic articles for the demonstration of 'Phantastic Objects'[7] and their effect is of growing importance to the understanding of financial and economic bubbles [2]. As an example of the application of social data to the social sciences, it is a logical next step to consider Phantastic Objects in every-day social media data. Such analyses could lead to new findings in the realm of 'meme' propagation, with a better understanding of why sentiment bubbles occur.

To achieve this, a manageable database of social media data is required, and with its data analytics capabilities, SocialSTORM is able to provide this stepping stone.

4.2 Business intelligence

A growing number of services provide business intelligence to firms seeking to monitor the sentiment around their company's name on the internet. Furthermore, sentiments around products influence future product design, and real-time analysis of the world's 'mood' on a brand or company dictates the success of digital marketing endeavors.

However, a particular aspect of digital business intelligence is currently in its infancy. It is becoming increasingly apparent that the instant global sentiment around a firm or its products can be used as a predictor of that firm's future performance. Whilst previously establishing such relationships was only possible after the time-consuming analysis of professionally-written publications on a company's performance or its products, it is now clear that public 'groupthink' opinion is just as relevant. The analysis of social media data in a near-instant capacity is thus the methodology needed, and SocialSTORM can facilitate this process.

[7] Stemming from the word 'phantasy', this term is used in the sense of meaning an imaginary scene in which the inventor of the phantasy is a protagonist in the process of having his or her latent (unconscious) wishes fulfilled [1].

4.3 Advanced prediction modeling

The prediction and/or estimation of macroeconomic variables such as consumer confidence, unemployment, and inflation are of great importance to both policy makers and investors. Current methodologies of observing such variables are highly limited. Not only are they survey-based, meaning that the data-accumulation and analysis process is extremely time and cost intensive, but the results are often cited in literature as being simply inaccurate. Much value therefore lies in the timely prediction of such macroeconomic variables, and it has already been shown that the analysis of sentiment derived from search engine data can be used as a predictor for financial and economic variables [3].

Thus, it is of interest to explore the effect of sentiment variation of social media data on macroeconomic variables such as those mentioned above. With SocialSTORM's capacity to monitor both historic and live data via the use of custom-written models, such analyses may now be feasible.

5 Future work

SocialSTORM is in constant development, and the following additional features are in the process of being implemented:

Sentiment classification – A series of in-built machine-learning packages are being developed which will allow for the sentiment analysis of the text stored in SocialSTORM's database. This analysis will allow users to quantitatively rank social media text based on emotions such as anger, anxiety, happiness and sadness.

Esper – This is a complex event processing package, which allows for the high-speed processing of large volumes of events in real-time[8]. The integration of this package will implement SocialSTORM's sentiment classification algorithms in real-time, before messages are stored in the database; and will also improve the system's overall performance when simulation models are operating on live data.

Advanced Visualization Suite – The current Web interface is being improved and expanded to allow for the customization of the visualization of the output data produced by the simulation models.

Integration with ATRADE – Upon completion, this functionality will allow SocialSTORM's users to run models that can simultaneously evaluate financial data from the ATRADE platform, as well as its native social media data.

[8] http://esper.codehaus.org/

6 Conclusion

UCL's SocialSTORM platform is a data mining and analytics engine that can provide access and customized monitoring capabilities for aggregated social media. Being a non-commercial product, the platform offers researchers a facility to monitor and evaluate a rich and yet often fleeing data source. The platform complements existing commercial products which offer similar capabilities, but are not primarily targeted at the wider academic community. SocialSTORM's customizable nature also allows for integration with local software to support research in Computational Social Science.

7 References

[1] J. Laplanche, J. Pontalis The language of psychoanalysis. Nicholson-Smith D, translator. New York, NY, London: Norton and Hogarth, 1973.

[2] D. Tuckett, R. Taffler, "Phantastic objects and the financial market's sense of reality: a psychoanalytic contribution to the understanding of stock market instability", Int. J. Psychoanal., vol. 89, 389–412, Jun 2008.

[3] H. Mao, S. Counts, J. Bollen, "Predicting Financial Markets: Comparing Survey, News, Twitter and Search Engine Data (Periodical style–Submitted for publication)", eprint arXiv:1112.1051, submitted for publication, Dec. 2011.

SESSION

NOVEL APPLICATIONS AND ALGORITHMS

Chair(s)

Dr. Robert Stahlbock

A Comparative Study of Predicting User Preference using Evolutionary Clustering Algorithm

Chhavi Rana[1] and Sanjay Kumar Jain[2]
[1] University Institute of Engineering and Technology, MDUniversity,Rohtak,Haryana, 124001, India.
email: chhavi1jan@yahoo.com
[2] Department of Computer Science Engineering, National Institute of Technology, Kurukshetra, Haryana, 136119, India.
email: skjnith@yahoocom

Abstract - *Recommender systems are the tools that predict user preferences and thus help a naïve user in finding useful information on the world wide web. They have become a necessary agent in the information bombardment arena of World Wide Web. A number of algorithms are implemented to predict the preference of user and thereby give them recommendation. Majority of these algorithm use data mining techniques. In this paper, we present a comparative analysis of various classification algorithm and there integration with various clustering algorithm that could effectively and accurately predict temporal changes in user preferences. The paper also presents a newly developed evolutionary clustering approach and its comparative analysis. Several experiments were conducted using these algorithms based upon various parameters using WEKA (Waikato Environment for Knowledge Analysis), a Data Mining tool. The results of the experiment show that integration of clustering and classification gives promising results with higher accuracy rate and lower error rates when compared over temporal parameters.*

Keywords: A Maximum of 6 Keywords

1 Introduction

The increased usage of web as a major medium of communication, business and entertainment lead to the rise of huge amount of stored web data. This data may contain valuable knowledge which could be beneficial for an organization as well as individual users. Data mining algorithms are generally used to deal with such kind of data. Data mining is the process of extracting new, useful and interesting information using a variety of techniques. These techniques are used by Recommender systems for making recommendations by means of knowledge extracted from the action and attributes of users [9].

A range of data mining techniques are developed for extracting useful information such as pattern recognition, clustering, association and classification [29]. The advantage of using data mining techniques for analysis includes objective and accurate results, and ease of application in routine tasks. Various machine learning researchers also emphasize the use of data mining techniques in information retrieval prominent of which is clustering and classification. Clustering is an unsupervised learning techniques which can identify and form groups of items/users based on similar features pattern in the data. On the other hand, classification is a supervised learning technique which find a set of prediction model that distinguish data class models for predicting the unlabelled class from the data. In this paper, we study the temporal evolution of user preferences through an integration of classification and clustering techniques and compare it with a benchmark baseline classification algorithm. In addition, the paper also outlines two other widely researched classification algorithm that have been applied to complex, high dimensional classification problems in the last few years and their feasibility in Rs application[13]. The proposed work will focus on finding useful information through time based analysis of e-commerce data to predict user preference and use temporal trends to increase the accuracy of predictions.

In this paper, weka [7] machine learning tool is used for performing evaluation using clustering and classification algorithm. The paper is organized as follows: Section 2 defined the background study of the specific domain taken up for analysis. Section 3 describe the proposed classification and clustering method to identify the class of uses preferences. Experiment results and performance evaluation are presented in Section 4 and finally Section 5 concludes the paper citing future work in this direction.

2 Background

Various classification and clustering algorithm are used in collaborative filtering and content filtering technique to built a model for predicting user preference and generating a recommendation list. However, to the best of our knowledge, no one has so far employed an integration of clustering & classification for finding temporal trends in user preference data. A number of researchers though have worked in this area, prominent of them are mentioned here.

The problem of finding temporal trends in the context of recommender system have been addressed recently in [32]

and [20]. In the area of recommender system research, most of the studies focus on increasing the accuracy of algorithm. For the prediction of user preference, most of the prediction model employ either collaborating filtering or content based filtering as a technique. It is being observed and analyzed by Koren [32] that after a time the accuracy of recommender system can only be increased by incorporating the temporal information. He also proposes a model that traces the time changing behavior throughout the life span of data and thus exploiting all relevant components which is in contrast with the earlier concept drift explorations where only single concept is traced.

On the other hand, Lathia, Hailes, & Capra [20] provide a different perspective, which is a system oriented approach different from [32] user preference model. He studies the effect of retraining CF algorithm every week as a time dependent prediction problem and proposes an adaptive temporal CF technique. Another approach that incorporated temporal information to achieve better recommendation accuracy is proposed by Queue et al. [28]. They combined the two dimensions involving temporal dynamics, one proposed by [27] involving product launch time and other by [31] that is based on rating time, together with implicit feedback data to construct a pseudo rating data.

A feature-based machine learning approach was proposed by Chu & park [30] to personalized recommendation that is capable of handling the cold-start issue effectively. They maintain profiles of content of interest, in which temporal characteristics of the content, e.g. popularity and freshness, are updated in realtime manner. Another recommender system named "Eigentaste 5.0: Constant-Time Adaptability Recommender System" was developed by Nathanson, Bitton, & Goldberg [26] that dynamically adapts the order that items are recommended by integrating user clustering with item clustering and monitoring item portfolio effects. Another group of researcher, Tang et al., [16] focused on dealing with the issue of scalability by applying a different kind of technique that scale down candidate sets by considering the temporal feature of items. Golbandi, Koren, & Lempel [19] proposed to use an adaptive bootstrapping process that elicits users to provide their opinions on certain carefully chosen items or categories while changing the questions with time adapting to user responses. In addition, Nasraoui et al., [21] studied evolving user profile scenarios and proposed a systematic validation methodology that allows for simulating various controlled user profile evolution scenarios and validating the studied recommendation strategies. In addition, Pessemier et al. [25] presents an empirical evidence that older consumption data has a negative influence on the recommendation accuracy in case of consumer centric RS. Chen et al. [24] also showed the time decaying effect of sequential pattern on the user preference within content based RS. Though several user prediction models mentioned previously provide very good predictions performance, the majority of these paper do not take into account the time. Most of these algorithms uses a generalized collaborative filtering technique with some modification and as such have almost similar prediction capability and performance. For a detail review, please see [9] and [10] as a starting references. The remaining algorithms focus on specific problem like scalability and cold start. The primary goal of our proposed work is to develop a more generic approach that updates and thereby improves existing state of art techniques.

3 Clustering and Classification

Cluster analysis is one of the most prominent methods for identifying classes amongst a group of objects and has been used as a tool in many fields such a as biology, finance and computer science [8]. Recent work by [3] and [15] shows that cluster analysis has the ability to group user using their preferences in rating data. In this paper we will go a step further and analysis how an integration of classification and clustering algorithm can be used to analysis and predict the preference of user based on its previous rating. We evaluate various clustering algorithm namely Kmeans, EM as well as our proposed clustering approach and their integration with benchmark baseline classification algorithm for collaborative filtering namely IBK, that to the best of our knowledge have not been previously applied to this problem. The clustering algorithm use an unsurpervised mechanism, where an unlabeled training data is grouped based on similarity [8]. This ability to group unlabelled training data is advantageous and offer some practical benefits over learning approach that requires labeled training data [12, 29]. The supervised learning approach cannot discover new application and only classify user/items for which it has labeled training data. Another advantage is when user ratings are being labeled, due to high accuracy of clusters, only a few items used need to be identified in order to label the cluster with a high degree of confidence.

Collaborative Filtering (CF) is the most widely used technique in recommender system for predicting user interest on unseen items by analyzing user's historical data [9]. Collaborative filtering uses a number of algorithms such as nearest neighbourhood, Bayesian beleifnets, clustering, matrix factorization and others [10]. In this paper, Item based Collaborative filtering algorithm based on an integration of clustering and classification are evaluated. Recommendation is treated as a classification problem and the classification problem is further simplified using clustering algorithm. This section reviews some other major classification algorithm.

3.1 Multiclassifier

Multiclassification systems have proved to be very promising tools to improve the accuracy of single classifiers when applied to complex, high dimensional classification problems in the last few years[13].Web system arena has hardly used one of the most reliable data mining models which are called multiclassifiers. Segrera & Moreno [23] presented a comparative study of different simple classifiers and multiclassifiers using a subset of dataset from MovieLens recommender system. They suggested the fact that multiclassifier takes more time in their execution which limits their use in the application areas which have relatively stable environment. Jiang, Shang and Liu proposed that good recommendation system should not only consider what the customer needs, but also ensures customer's contentment. For implementing this proposition, they formulated a rating classification model based on the customer's profile and feedback[33]. The standard classification models for recommender systems are reinvestigated by [14]. They

identified that the intrinsic autocorrelation structure and the absence of item re-occurrences (repeat buying) are those two properties. Furthermore, they suggested a solution by providing a generic framework for using any binary classifier for recommendation generation.

3.2 Ensemble Learning

Ensemble regression methods can be used for the prediction of missing ratings in recommender systems. Schclar et al. [1] studied how predicting the ratings can be formulated as a regression problem and they used an adaptation of an ensemble method, Adaboost for this task. Wu[18] also showed how matrix factorization method can be combined through ensemble learning with different parameter to give better results.

3.3 Clustering via classification

The integration of clustering and classification is being studied by very few researchers. Kumar [29] compared results of simple classification technique with the results of integration of clustering and classification technique, based upon various parameters using publicly available dataset named Iris. Erman, Arlitt, & Mahanti [8] demonstrated how traffic groups can be classified effectively through cluster analysis. An efficient anomaly based network intrusion detection model is build by using a novel classification via sequential information bottleneck (sIB) clustering algorithm[16].

4 Proposed Method

Evolutionary clustering is applied in many real world problems such as market segmentation, social network analysis, web mining and bioinformatics. Evolutionary clustering is often considered as an offshoot of Incremental clustering as well as methods that are used in clustering data streams. Though both of them are similar in the sense that they all deal with data that changes with time, but the difference is well explained by Shankar et al. [22]. Data stream clustering focuses on optimizing time and space constraints while evolutionary clustering is concerned about temporal smoothness. Similarly, Incremental clustering does not maintain relevancy to existing clustering as opposed to evolutionary clustering.

On one hand, a number of evolutionary algorithms for solving clustering problem have been proposed that treat clustering as NP hard problem and are based on optimization of some objective function [5]. On the other hand, Chakarbarti [4] is taking a completely different outlook. He implemented evolutionary clustering by building a framework termed as temporal smoothness that produces updated clusters from data coming at different time stamps. This method combines two conflicting objectives called snapshot quality and history cost. The clustering algorithm should tradeoff the advantage of maintaining a consistent clustering overtime termed as snapshot quality with the cost of deviating from an accurate representation of current data.

In this paper, a new optimizing approach named EVar to discover updated cluster in the dynamic environment using evolutionary approach is presented. User preferences are represented in the form of these updated clusters. The discovery of clusters that evolves with time and finding new preference in terms of updated clusters can be presented as an optimizing problem. The conflicting objectives named snapshot quality and history cost needs to be maximized and minimized simultaneously.

4.1 Problem formulation

The field of data mining is a combination of various learning algorithm. Clustering algorithm is one such category that is widely used in unsupervised learning. Recommender System, on the other hand uses a technique called collaborative filtering which gives recommendation by finding similar users and predicting new user preference based on their similarity. Clustering algorithm have been very efficiently applied in the process of collaborative filtering to find similar users by developing clusters of similar users or items for prediction. However, with changing user requirement, a new mechanism is required to give accurate recommendations. For implementing such recommendation system, a new method EVar is proposed here. EVar uses a framework of Evolutionary clustering introduced by Chakarbarti [4], which clusters data over a period of time. At each time stamp, a new cluster is produced by optimizing two competing parameter named snapshot quality and history cost. Snapshot quality refers to the quality of clusters formed and how accurately they depict data at the current time while history cost implies that the new cluster should not differ dramatically from the earlier one. This framework in fact focuses on smooth transition of cluster, which evolves over time by maximizing snapshot quality and minimizing history cost. The total quality of the sequence is defined as follows:

$$Sq(CtMt) - CpHc(Ct-1, Ct)$$

where

Sq (C_tM_t) return quality of cluster C_t at time t w.r.t input m

Hc (C_{t-1},C_t) return history cost of cluster C_t at time t w.r.t time t-1

Cp denotes the parameter for adjustment of the two objectives

We define Evolutionary Clusters as a group of items at a given time stamp t. Let T = {1,2,3. . . ,t } be a finite set of time stamps and I = {i_1,i_2 . . . , i_t} be a set of items or users arrived at different time stamps. Let C = (C_1, C_2, C_t) be set of clusters at different timestamps. Let C_1 be the cluster at

timestamp T_1. When a new item i_t is added at timestamp t or an old item changes its preference level and variance score, the EVar produces a new clustering C_t, which optimizes the quality of clusters. A cluster group C_t where $C_t = \{C_{t1}, C_{t2}, C_{t3} \ldots C_{tk}\}$ (here k depicts the number of cluster groups) is a group of items at any given time and the partitioning of items is such that each group gives the maximum similarity and minimum variance. This is achieved by clustering algorithm through defining their cost function depicting the quality of clusters.

Different functions are further proposed by different researchers to determine the snapshot quality and history cost. EVar clustering algorithm proposes a cost function which uses variance score to determine the snapshot cost in the temporal smoothness framework given by Chakarbarti[4] and further adopted by Kim and Han [11, 17]. As snapshot quality measures how well the cluster represents the data at time t, we have defined this measure called variance score, which minimizes difference within the items in a cluster and maximizes the similarity. The implementation of variance introduced in [6] has proved very effective in determining snapshot quality. Variance score is the difference between the ratings of items in a particular cluster at a given point of time. Greater the value of the variance score, lower will be snapshot quality.

$$Sq = \sum (1 - VScore(M_t, t))$$

where

Sq is the snapshot quality

VScore is the variance score at time t w.r.t M_t

$$\text{VScore}(u, i) = \sum_{u' \in k(u)} (R(u', i) - R(i))^2 / K$$

where

VScore denotes variance of K neighbor rating for item i

K (u) denotes k neighbors of user u who rated item I and has the highest similarity sim (u, u') to user u.

R (u) denotes the average rating of user u

R (i) denotes the average rating of all K neighbors on item i

I (u, u') is the set of all items rated by both user u and u'

The history cost is defined using traditional entropy measures NMI [2].

The normalized mutual information NMI (A, B) is defined as:

$$NMI(A, B) = -2 \sum_{i=1}^{CA} \sum_{j=1}^{CB} C_{ij} \log(C_{ij} N / C_i . C_j) /$$

$$\sum_{i=1}^{CA} Ci. \log(Ci.N) + \sum_{j=1}^{Cb} Cj. \log(Cj.N)$$

Where C_A (C_B) is the number of groups in the partitioning A (B), Ci. (C_j) is the sum of the elements of C in row i (column j), and N is the number of nodes. If A = B, NMI (A, B) = 1. If A and B are completely different, NMI (A, B) = 0. Thus, our second objective at a generic time step t is to maximize NMI (C_t, C_{t-1}). Recall that U = {one, 2, 3 . . ., n} is the universe of objects to be clustered. At each timestamp t where $1 \leq t \leq T$, a new set of data arrives to be clustered. We assume that this data can be represented as an $n \times n$ matrix M_t that expresses the relationship between each pair of data objects. The relationship expressed by M_t can be either based on similarity or based on distance depending on the requirements of the particular underlying algorithm. If the algorithm requires similarities (resp., distances), we will write sim (i, j, t) (resp., dist (i, j, t)) to represent the similarity (resp., distance) between objects i and j at time t. At each timestamp t, an online evolutionary clustering algorithm is presented with a new matrix M_t, either sim (・ , ・ , t) or dist (・ , ・ , t), and must produce C_t, the clustering for time t, based on the new matrix by calculating snapshot quality and history cost so far.

5 Experiment

The experiments were carried out using the Java open-source program named Weka. Weka provides environment for comparing learning algorithms, graphical user interface, comprehensive set of data pre-processing tools, learning algorithms and evaluation methods [5]. Furthermore Weka provides implementation of Clustering and Classification that we will be using for our comparative analysis. Moreover, we have also implemented our proposed evolutionary clustering algorithm in Weka to compare it with other algorithms. As part of our experiment, we used the classification algorithm IBK which is an open source Java implementation of the Nearest neighbor algorithm in Weka.

5.1 Testing Methodology

Three clustering analysis methods and three classification methods have been used to examine the efficiency of the including temporal features for predicting recommendations in a recommender system. For clustering, we have used SimpleKMeans and EM (Expectation Maximization) algorithms, both of them supported in Weka system. The third method is our proposed approach named evolutionary clustering which is implemented in Weka as well that include the use of temporal information. The three clustering algorithm are integrated with an algorithm named IBK which is a benchmark baseline collaborative filtering used in many existing recommender system. The effect of integration is then studied with baseline algorithm. Furthermore two other classification algorithm namely ensemble learning methods and multiclassifiers are also compared to study the comparison between simple classification and its integration with clustering.

5.2 Dataset

In this experiment, we present a comparative study of classification technique of data mining with an integration of

clustering on various parameters using a subset of Movielens dataset containing 3000 instances and 4 attributes. The dataset consists of three files namely user data, movie data, rating data. The dataset comprising of all the instances was divided into 80%-20% training and test set, and then the training set has been used to learn the model but the test set is used to assess the quality of the final model, i.e., to compute the Mean absolute error (MAE) and Root mean square error (RMSE).

5.3 Experiment Result

The overall effectiveness of the data mining algorithms is calculated using overall accuracy. Generally, MAE (Mean Absolute Error) is used as the standard metric for evaluating CF algorithms. The MAE is sensitive to the number of folds used for cross-validation, since the more items are used for training, the better the prediction. For this reason, MAE and other metrics are meaningful only within the context of this paper, or when fairly compared with measures from a framework using the same tests.

5.3.1 Integration of classification with clustering

The results of the experiment show that integration of clustering and classification over temporal parameter gives promising results with higher accuracy rate and lower error rates when compared with the standard baseline classification model. Particularly, our proposed clustering algorithm gives the best statistics and proves to depict highest accuracy when time is taken into account (Table 1). An experiment measuring the accuracy of IBK classifier based on MAE (Mean absolute error) and RMSE (Root mean square error), RRAE (Root Relative absolute error), RRSE (Relative Root square error), and its integration with Kmeans, EM and EVar clustering algorithms is shown in Table 1.

It may be observed from Table 1 that the error rate of binary classifier IBK with EVar Clusterer is lowest i.e. 0.0012 in comparison with IBK classifier without clusterer which is most desirable as well as in comparision to other clustering algorithm.

- Accuracy of IBK classifier with EVar clusterer is high i.e. 99.4424% (Table 1), which is highly required.
- RAE (TPR) of clusters (results of integration of classification and clustering technique) is lower than that of classification techniques (Table 1).
- In an ideal world we want the MAE to be zero. Considering results presented in Table 1, MAE is lowest in case of integration of clustering and classification technique. In other words, it is closed to zero in comparison with simple classification technique with IBK classifier.

According to the experiments and result analysis presented in this paper, it is observed that an integration of classification and clustering technique is better to classify datasets with better accuracy.

Table 1. Classifier Comparision

Parameters	IBK	IBK+ EM	IBK+ Kmeans	IBK+ EVar
MAE	12052704.1096	0.0042	0.0014	0.0012
RMSE	20311725.6681	0.0042	0.0014	0.0012
RRAE	84.7866 %	2.3819 %	0.6789 %	0.5576 %
RRSE	115.2278 %	1.405 %	0.4822 %	0.4583 %
Instances	1054	122	358	550

5.3.2 Individual Classification Algorithms

Furthermore, two more classification algorithm are compared with baseline classifier named IBK. Firstly, the behaviour of Bagging and Stacking with our previously tested classification algorithm, IBK and their integration with clustering were analyzed in order to determine if multiclassifier could be used to increase the accuracy while predicting user preferences.

Building and evaluation times of the individual algorithms are short in relation to Bagging and stacking as showed in (Table 2). The simple IBK classifier increased its execution time significantly when bagging and stacking are applied. On the other hand, there is no significant difference in the time of classifier that are integrated with clustering algorithm. Following the analysis of Table 3, we can discard IBK classifier, which presented the higher relative absolute error in comparison to other algorithms. Moreover, the ensemble methods Bagging and stacking that used IBK also does not showed very sharp improvement in relative absolute error values in case of individual classifier. In WEKA for this case study, the integration of classifier with clustering by different algorithms showed some improvement through bagging (Table 3). Hence, the use of multiclassifiers, that increase the model building and evaluation time only to improve in hundredth to most of the error values could not be justified. On the other hand, the hybrid method that integrate our proposed EVar clustering algorithm with baseline classifier IBK decreased in around 5% (7.7334 %) the relative error by the individual classifier (1.5632%) with IBK through bagging while it was decreased only 3% (8.7866 %) by the individual classifier (5.5546%). In comparision, the other two clustering algorithm (Kmeans and EM) were used as base clusters to build hybrid

Table 2. Building and Evaluation Time(in sec)

Classification algorithm	Simple classifier	Bagging	Stacking
Nearest Neighbour (IBK)	2	6	40
IBK+ EM	4	10	50

IBK+ Kmeans	4	11	55
IBK+ EVar	4	11	47

Table 3. Relative absolute Error values

Classification algorithm	Simple classifier	Bagging	Stacking
Nearest Neighbour (IBK)	8.7866 %	7.7334 %	99.8942 %
IBK+ Kmeans	3.6789 %	2.0794 %	91.1132 %
IBK+ EM	7.3819 %	6.9521 %	96.0331 %
IBK+ EVar	5.5546%	1.5632 %	98.771%

Table 4. Overall statistics

Classification technique	Clustering technique	Accuracy	TA RMSE	Time
IBK	Kmeans	0.6789 %	0.0044	55
IBK	EM	2.3819 %	0.0062	50
IBK	EVAR	0.1562%	0.0014	47
IBK	None	84.7866 %	10215785.5691	40
Multiclassifier bagging	None	80.334 %	19987339.5375	70
Ensemble selection	None	55.931 %	14260705.4617	60

models and the metaclassifier applied was the nearest neighbor algorithm (IBK). In spite of the error value improvement by our hybrid integration, the obtained values by the individual classifier were also comparable. In addition, the execution time of our model was higher than that of the individual classifier, which discouraged the use of multiclassifiers based on IBK to be used for predicting user preferences.

The Overall experimental results of our approach as presented in Table 4. In this study, the accuracy of three data mining techniques is compared. The goal is to have high accuracy, besides low RAE metrics. Although these metrics are used more often in the field of information retrieval and can be easily derived from the confusion metrics. Temporal trends cannot be determined by them. For analyzing temporal accuracy, a Time averaged RMSE parameter as suggested by Neal lathia [20] is calculated which is discussed in the next section. As can be seen in Table 4, Kmeans and EM have comparable performances when integrated with IBK. The results clearly show that the error rate of these algorithm (0.675-2.38%) is much higher than the error rate of our approach (~0.15%) that combined EVAR clustering before classification. It may be worth noting that the computation times of the algorithms Kmeans, EM and EVAr (on an Intel core2 duo machine) were in the ranges of 55 sec, 50 sec and 47 sec respectively which are comparable. The third and the most important parameter that gives the overall trend using temporal parameter also validated previous result. The values of TA RMSE of our proposed approach were the lowest among the compared approaches. According to the experiments and result analysis presented in this paper, it is observed that an integration of classification and clustering technique is better to classify datasets with better accuracy and less temporal error. On the other hand, multiclassifiers are sensitive to the data quality from the web. Its application in predicting user preference must be considered if the employed time in the model building is not extended.

Conclusion

The difficulty to identify temporal changes in user preferences has shifted focus on the usage of clustering algorithm for detecting such effect. Cluster analysis being an unsurprised learning algorithm is a good condition for predicted similar groups of user having similar data. This detection could further be classified using a model that generated accurate model prediction user rating. This paper examine the integration of classification and clustering technique for continuous prediction of user preferences over a period of time and accurately determining his interests. Users rating are grouped into similar preference using clusters and they are further modeled using a classification algorithm. The result of integrating clustering with classification have given promising results. A number of approaches are experimentally analyzed involving clustering and classification algorithms. In conclusion, the field of recommender system is going to be more actively researched with the rise of data on the world wide web.

References

[1] A. Schclar, A. Tsikinovsky, L. Rokach, A. Meisels, & L. Antwarg, "Ensemble methods for improving the performance of neighborhood-based collaborative filtering", *Proceedings of the third ACM RecSys 09*, pp. 261-264, 2009.

[2] A. Strehl and J. Ghosh, "Cluster Ensembles a knowledge reuse framework for combining partitions", *The Journal of Machine Learning Research*, Vol. 3, Cambridge, MA, USA, pp. 583-617, 2002.

[3] B. M. Sarwar, G. Karypis, J. Konstan, & J. Riedl, "Recommender Systems for Large-scale E-Commerce : Scalable Neighborhood Formation Using Clustering". *Communications*, 2002.

[4] D . Chakrabarti, R. Kumar, & A. Tomkins, "Evolutionary clustering", *Proceedings of the 12th ACM KDD 06*, Vol. 8, No. 4, pp. 554-661, 2006.

[5] E. R . Hruschka, R. J. G. B . Campello, A. A . Freitas, and A. C. P. L. F. De Carvalho, "A survey of evolutionary algorithms for clustering", *IEEE Transactions on Systems Man and*

Cybernetic Part C Applications and Reviews, Vol. 39, No. 2, pp.133-155, 2009.

[6] G. Adomavicius, & Y. Kwon, "Overcoming Accuracy-Diversity Tradeoff in Recommender Systems: A Variance-Based Approach", *Proceedings of WITS*, Vol. 8, Paris, France, 2008.

[7] G. Holmes, A. Donkin, I. H. Witten, "WEKA a machine learning workbench", *Proceeding second Australia and New Zeeland Conference on Intelligent Information System*, Brisbane, Australia, pp.357-361, 1994.

[8] J. Erman, M. Arlitt, & A. Mahanti, "Traffic classification using clustering algorithms", *Proceedings of the 2006 SIGCOMM workshop on Mining network data MineNet 06*, *1*(Ldm), pp. 281-286, 2006.

[9] J. B. Schafer, "The application of data-mining to recommender systems", In J. Wang (Ed.), Encyclopedia of data warehousing and mining, Hershey, PA. Idea Group, pp. 44-48, 2005.

[10] J. S. Breese, D. Heckerman, & C. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering", *Proceedings of the 14th conference on Uncertainty in Artificial Intelligence*, Vol. 461, No. 8, pp. 43–52. San Francisco, CA, 1998.

[11] K. Kim, and R. I. B. Mckay, "Multiobjective Evolutionary Algorithms for Dynamic Social Network Clustering", *Proceedings of the 12th annual conference on Genetic and Evolutionary Computation GECCO2010*, pp. 1179-1186, 2010.

[12] K. Lakiotaki, N.F. Matsatsinis, & A. Tsoukiàs, "Multi-Criteria User Modeling in Recommender Systems", *Decision Support Systems,* Vol. 26, pp. 64–76, 2011.

[13] L. Kuncheva, "Combining Pattern Classifiers: Methods and Algorithms", *Wiley*, 2004.

[14] L. Schmidt-thieme, "Compound Classification Models for Recommender Systems", IEEE *Work,* pp. 378-385, 2005.

[15] M.C. Pham, Y. Cao, R. Klamma, & M. A. Jarke, "Clustering Approach for Collaborative Filtering Recommendation Using Social Network Analysis", *Journal Of Universal Computer Science,* Vol. 17, pp. 1-21, 2010.

[16] M. Panda, & M. R. Patra, "A Novel Classification via Clustering Method for Anomaly Based Network Intrusion Detection System" *International Journal*, Vol. 2, No. 1, pp. 1-6, 2009.

[17] M.-S. Kim, & J. Han, " A Particle-and-Density Based Evolutionary Clustering Method for Dynamic Networks". *VLDB*, Vol. 2, No. 1, pp. 622–633, 2009.

[18] M. Wu, Collaborative Filtering via Ensembles of Matrix Factorizations, "*Biological Cybernetics",* pp. 43-47, 2007.

[19] N. Golbandi, Y. Koren, & R. Lempel, "Adaptive Bootstrapping of Recommender Systems Using Decision trees", WSDM, Current, pp. 595-604, 2011.

[20] N. Lathia, S. Hailes, & L. Capra, "Temporal collaborative filtering with adaptive neighbourhoods", *Proceedings of the 32nd international ACM SIGIR conference on Research and development ininformation retrieval SIGIR 09*, pp. 796-797, 2009.

[21] O. Nasraoui, J. Cerwinske, C. Rojas, & F. Gonzalez, "Performance of Recommendation Systems in Dynamic Streaming Environments", Proc. of SDM – SIAM International Conference on Data Mining. Minneapolis MI, 2007.

[22] R. Shankar, G. V. R Kiran, and V. Pudi, "Evolutionary clustering using frequent itemsets", *Proceedings of the First International Workshop on Novel Data Stream Pattern Mining Techniques StreamKDD 10,* ACM Press, pp. 25-30, 2010.

[23] S. Segrera & M.N. Moreno. "An experimental comparative study of web mining methods for recommender systems", In *Proceedings of the 6th WSEAS International Conference on Distance Learning and Web Engineering* (DIWED'06), Sebastiano Impedovo, Damir Kalpic, and Zoran Stjepanovic (Eds.). World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, pp. 56-61, 2006.

[24] T. Chen, W. L. Han, H. D. Wang, Y.X. Zhou, B. Xu, & B. Y. Zang, "Content recommendation system based on private dynamic user profile", International Conference on Machine Learning and Cybernetics, pp. 2112-2118, 2007.

[25] T. D. Pessemier, T. Deryckere, T. Deryckere, & L. Martens, "Time dependency of data quality for collaborative filtering algorithms", Proceedings of the fourth ACM conference on Recommender systems, Barcelona, Spain, pp. 281–284, 2010.

[26] T. Nathanson, E. Bitton, K. Goldberg, "Eigentaste 5.0: constant-time adaptability in a recommender system using item clustering", *ACM RecSys*. pp. 149-152, 2007.

[27] T. Tang, Winoto, P. and Keith, C. Chan, C.: Scaling Down Candidate Sets Based on the Temporal Feature of Items for Improved Hybrid Recommendations. Intelligent Technique of Web Personalization. LNCS vol. 3169. pp.169-186, 2005.

[28] T. Queue, Y. Park, & Y. T. Park, "A time-based approach to effective recommender systems using implicit feedback. Expert Systems with Applications", Vol. 34, No.4, pp. 3055-3062, 2008.

[29] V. Kumar, "Knowledge discovery from database Using an integration of clustering and classification", *IJACSA International Journal of Advanced Computer Science and Applications*, Vol. *2, No.* 3, pp. 29-33, 2011.

[30] W. Chu, S. T. Park, "Personalized Recommendation on Dynamic Content Using Predictive Bilinear Models", 18th International WWW Conference. Madrid, Spain, pp 691-706, 2009.

[31] Y. Ding, X. Li, & M. E. Orlowska, "Recency based collaborative filtering", *Proceedings of the 17th Australasian Database Conference, Australian Computer Society, Inc.*, pp. 49, 99–107, 2006.

[32] Y. Koren, "Collaborative filtering with temporal dynamics", *Communication of the ACM*, Vol. 53, No. 4, pp. 89-97, 2010.

[33] Y. Jiang, J. Shang, & Y. Liu, "Maximizing customer satisfaction through an online recommendation system: A novel associative classification model", *Decision Support Systems*, Elsevier, Vol. 48, No. 3, pp. 470-479, 2010.

Parallelization Strategies for Distributed Non Negative Matrix Factorization

Ahmed Nagy
IMT Institute for Advanced Studies Lucca
Italy
ahmed.nagy@imtlucca.it

Massimo Coppola
ISTI - CNR Pisa
Italy
massimo.coppola@isti.cnr.it

Nicola Tonellotto
ISTI - CNR Pisa
Italy
nicola.tonellotto@isti.cnr.it

ABSTRACT

Dimensionality reduction and clustering have been the subject of intense research efforts over the past few years [2]. They offer an approach of knowledge extraction from huge amounts of data. Although some of these techniques are effective at achieving lower data dimensions, very few focused on scaling the techniques to tackle data sets that might not fit into memory. Non negative matrix factorization is (NMF) one of the effective techniques that can be used to achieve dimensionality reduction, missing data prediction and clustering. NMF has been parallelized through shared memory and distributed memory. Our contribution lies in reaching a higher level of parallelism through proposing a new block division technique on a *hadoop* framework. Furthermore, we use the block-based technique to design an enhanced cascaded NMF [6]. We compare the division techniques that we propose, block-based and cascaded over block-based division to the column-based technique which exists in the literature [12]. The block-based technique performs 18% percent faster than the column based. It achieves higher convergence value than the cascaded technique by 23%

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Storage and Retrieval—*Clustering*

Keywords

Knowledge Extraction, Data mining, Non negative matrix factorization, Dimensionality reduction, Missing data prediction.

1. INTRODUCTION

The explosion of daily data generated by the Web urges devising new ways to decrease the dimensionality and deal with missing data [10]. After Lee published his paper [9], Non Negative Matrix Factorization (NMF) has gained great importance. Matrices provide a compact means to describe relations among entities. For example relations between pages and tags can be easily described in a matrix format. However, this leads to big matrices that suffer from high dimensionality. Decomposing the matrices can be used to describe the big matrix in terms of two lower dimensional matrices. NMF aims at decomposing a matrix $V^{n\times m}$ into two lower dimensional matrices $W^{n\times k}$ and $H^{k\times m}$, where k is the dimensionality reduction factor, W is the weight matrix and H is the feature matrix. NMF is a decomposition technique very similar to Principal Component Analysis (PCA) and Singular Value Decomposition (SVD). However, it restricts matrices to non negative [2] which is an important feature for Web-based observations. The non negativity leads to computational savings especially in image processing, text mining, medical applications, decision making and disaster management since the observations are specified by numbers greater than or equal zero. As a result, there is no need to map the negative values to a non negative domain. To decompose the matrix iterative updates are used to compute W and H [9]. Multiplicative or additive updates can be used to decompose the observation matrix in terms of lower dimensional matrices. Multiplicative updates provide faster convergence [2, 12]; as a result, we will focus on multiplicative updates.

The contribution of our work stems from extending the non negative matrix factorization technique to scale on a cloud computing platform, *hadoop* [1]. The novelty of our work lies in two main areas. First, we consider matrices that might not fully fit into memory as opposed to methods that make the assumption that the matrices can fully fit in memory. Our second, contribution is reaching higher level of parallelism through proposing the block division technique on a *hadoop* framework. We use the block based technique to propose an enhanced cascaded NMF. We apply the map-reduce paradigm to calculate the block, column, row-based division, and the cascaded factorization.

Although NMF has several advantages when it comes to classifying and unleashing latent meanings, parallel implementations are still at infancy, especially a map-reduce implementation that could scale well. The method has not been fully explored. A distributed implementation is very crucial to develop scalable methods to use NMF. By developing the distributed non negative matrix factorization, we aim at analyzing different factors that could affect the factorization. The remainder of this paper is organized as follows. Section 2 introduces related work. Section 3 presents our approach for calculating the non negative matrix factor-

ization. The experimental setting and results are described in section 4. Finally we present our conclusions and future work in section 5.

2. RELATED WORK

NMF provides a non unique approximate factorization which suits Web scale data, where the amounts of data are huge and there is a need for fast processing. Furthermore, providing scalable solutions fast solutions are preferred to exact slow ones. When the amount of data increases, NMF provides better results for ranking the relevant resources [13, 14, 16, 18]. NMF provides better latent semantic indexing and prediction than SVD in 55% of the queries [4]. NMF does not require further processing for the matrix after factorization to cluster or categorize the objects [8, 17]. In [6], NMF is shown to have higher levels of convergence when provided with more iterations as compared to SVD.

One of the techniques that aimed to parallelize NMF was cascading [4]. The cascading technique gave promising speedup results and low inter process communication. The main idea behind the cascaded non negative matrix factorization is distributing the matrix on the stations (computational nodes in *hadoop*) then run NMF on each sub matrix and collect the result. The technique can be described as a data division approach to reach higher speedup. The process can be divided into two stages the parallel stage, where the NMF is calculated for every sub matrix, followed by the serial stage, where all the sub matrices are gathered. After dividing the input matrix into parts, NMF is run on each part alone. Later the factorizations of the sub matrices are gathered in the serial stage and another factorization stage is run on the results of the previous stage. The serial stage is used to build and consolidate the factorization of the input matrix. Though the scheme scales well, there were two caveats that the researchers did not completely address. The first one is the effect of dividing the matrix and carrying out the factorization on the convergence quality. The other one is a bottle neck and an explosion of the size of intermediate data resulting from the first cascaded level. Since every station factorizes part of the fed matrix, the result is a weight and a feature matrix for every station. As a result, the size of the weight matrix after consolidation increases by a factor equivalent to the total number of the stations. Our approach addresses the two former points and reveals the different compromises. In addition, we use our block division technique to address the bottle neck in the second cascading stage.

Another interesting attempt to parallelize NMF was presented in [12]. The scheme was built on *HAMA*, a distributed computing framework based on a bulk synchronous parallel model and uses *Hbase* as a way to store data [15]. *Hbase* is a distributed column-oriented store model [3]. The approach presented using *HAMA* depends on dividing the data in a column-wise order then using *Hbase* as a way to store the different portions of the matrices. The parallelization technique presented depends on delegating the parallelization to *HAMA*. The paper compared NMF with SVD and showed faster convergence for NMF by approximately 22%. We compare our block-based division approach with the column-based division proposed in [12]

Another parallelization approach was presented in [13] where the matrix V was divided into two parts and factorized in a very similar manner to the cascaded scheme. After the division phase, each portion is fed to factorized independently. Afterwards the factorization of the two portions was consolidated to form the factorization for the whole data. The method yielded a speed up of 180% approximately compared to the non parallel version. However, no measurements regarding the convergence of the method were provided.

3. PARALLEL NMF DESCRIPTION

3.1 Updating Equations

In order to calculate W and H, where W is the weight matrix and H is the feature matrix, equations 1 and 2 are used iteratively for the total number of iterations specified for the factorization module, introduced by [9]. Equations 1 and 2 assume that V, W and H exist on the same machine with no need to carry out any matrix division or distribution.

$$H \leftarrow H \frac{W^T V}{W^T W H} \tag{1}$$

$$W \leftarrow W \frac{V H}{W H H^T} \tag{2}$$

One of the critical design decisions is the way the observation matrix is split and distributed on the nodes. There are three main ways to split the matrix: block, column and row. In this paper we will illustrate the block-based due to the limited space though the other schemes were implemented. Figure 1 shows the way the matrix V is divided. Equations 3 and 4 show how the sub matrices of W and H are updated. A block is described by two indices the row and the column where we refer to the row as Q and to the column as P. The equations for updating the block-based scheme can be described as

$$H^P \leftarrow H^P \frac{\sum_{q=1}^{Q} (W^q)^T V^{q,P}}{(\sum_{q=1}^{Q} (W^q)^T W^q) H^P} \tag{3}$$

$$W^Q \leftarrow W^Q \frac{\sum_{p=1}^{P} V^{Q,p} (H^p)^T}{W^Q (\sum_{p=1}^{P} H^p (H^p)^T)} \tag{4}$$

The data dependency can be described as the Q intermediate results $(W^Q)^T \cdot V^{Q,P}$ and $(W^Q)^T \cdot W^Q$. Equations 3, 4 and Figure 1 show how the observation matrix V, weight matrix W and feature matrix H are divided. For example, in order to calculate H, $Q-1$ sub matrices of dimensions $K \cdot (N/P)$ and $Q-1$ results of dimensions K^2 should be sent for the nodes. From the above equations it can be observed that the calculation of H^P depends on the corresponding V^P, W and the current estimate of H^P to be distributed for the computation. In the block-based scheme the observation matrix V is split in both directions M and N, as shown in Figure 1. The input is split as P parts along the N dimension and into Q parts along the M direction, which results in having $P \cdot Q$ chunks or sub matrices. The chunks are defined by two indices P and Q. On the other hand, the matrix W is split into $W^1 ... W^Q$ and H into $H^1 ... H^P$.

3.2 Block-based Distributed NMF

Algorithm 1 shows the pseudo code for the distributed NMF factorization. The module receives a matrix V, dimension k

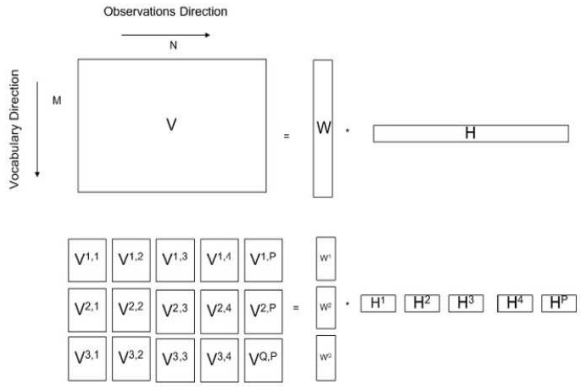


Figure 1: Splitting observations and variables, block scheme.

to factorize the matrix, the way the matrix will be divided, either block-wise, row-wise or column-wise and finally the total number of iterations. Lines 3 and 4 call the initialize method for W and H which carries out a random initialization for both matrices. The matrices are then mapped on the stations. The loop updates the matrix W and H according to equations 1 and 2 for the number of *TotalIterations* specified.

Algorithm 1 Distributed non negative matrix factorization

1: **Input**:
- **Matrix** V
- **Dimension** k
- **Type** *Division*
- **int** *TotalIterations*

2: **Output**:
- W Weight matrix
- H Feature matrix

3: Initialize(W);
4: Initialize(H);
5: Divide (V, *Division*, *Size*);
6: Map(V, W, H);
7: **for** $i = 0$; $i < TotalIterations$; $i++$ **do**
8: Update(W);
9: Update(H);
10: **end for**
11: $[W, H]$=Reduce();

3.3 Cascaded NMF Description

Algorithm 2 illustrates the cascaded NMF protocol. The module takes a matrix V to factorize, dimension k and number of splits. The first factorization stage (layer) initializes W and H exactly as the basic NMF described by Lee [10]. Lines 5-7 divide the matrices V, W and H into the number of splits. They are collected to be factorized later, every V split independently. There is no communication among the nodes while the splits are factorized. After factorizing the splits of V intermediate results of W and H are gathered in line 10 where CalculateNMF returns two matrices for every call. The resulting matrices are W^r, H^r where r is the id of the split where $0 \leq r \leq Splits$. Lines 13 and 14 consolidate the sub matrices of W and H in order to be sent for the second facorization cascaded stage (layer) at line 15. After

Algorithm 2 Distributed cascaded non negative matrix factorization

1: **Input**:
- **Matrix** V
- **Dimension** k
- **int** *Splits*
- **int** *TotalIterations*

2: **Output**:
- W Weight matrix
- H Feature matrix

3: $Initialize(W)$;
4: $Initialize(H)$;
5: $[V, ...V^{Splits}] = Divide(V, Splits)$;
6: $[W, ...W^{Splits}] = Divide(W, Splits)$;
7: $[H, ...H^{Splits}] = Divide(H, Splits)$;
8: **for** r=0; r < Numberofnodes; r++ **do**
9: **for** i=0; i < TotalIterations; i++ **do**
10: $[W^r, H^r]$=CalculateNMF(V^r,k);
11: **end for**
12: **end for**
13: $[H^*] = [H^0, ...H^r]$;
14: $[W^+] = [W^0, ...W^r]$;
15: $[W, H^\#] = DistributedNMF([W^+], k, block, 1)$;
16: $[H] = Multiply([H^*], [H^\#])$;
17: $return$ $[W, H]$;

collecting the intermediate Weight matrices, W^+ and feature matrices H^*, the second cascading phase starts. The second cascading phase starts at lines 15-16, where W^+ and H^* act as an input. It is clear that the number of Weight matrices W is equivalent to the number of splits which results in an inflation in the amount of the intermediate data that is needed to complete the factorization. This might put a limitation on the ability of cascading algorithm to scale. However, we further extended the Cascading algorithm by using our distributed NMF to carry out the factorization in the intermediate phase. The other limitation that is inherent in the cascading scheme stems from the way the data is factorized. In the first cascading layer there is no communication among the nodes.

The no communication approach has a benefit of decreasing the amount of data needed to be shuffled, and letting the factorization schemes able to proceed without waiting for results from each other. On the other hand, it affects the global values of the factorization as we will illustrate in the prediction experiment in the current section. After the first cascading phase which results in independent factorization for the sub matrices of V, completed at line 10, the second cascading layer starts by factorizing the consolidated weight matrices that were gathered from the first phase.

4. EXPERIMENTAL ENVIRONMENT

In this section we describe our evaluation environment for the calculation of non negative matrix factorization. As we

mentioned in the previous section that our aim is to calculate the weight matrix W and the feature matrix H for a given matrix V. We present detailed experiments that reveal the different performance characteristics of the factorization schemes developed, followed by an analysis for the results. We ran the experiments on an eight node cluster each has 1 GB ram, a hard disk of 70 GB and a 2 GHz $Xeon$ processor. We developed the modules on *hadoop* 0.21. and tuned its configuration by disabling speculation [5, 11].

The matrices that are fed to the module were generated by a random matrix generator that follows a $Gaussian$ distribution and a $Zipf$ distribution, home developed. We chose to test our module on both $Gaussian$ and $Zipf$ distribution since they are relevant to the problem we are addressing which is text mining. We focus on the results of the $Gaussian$ distribution due to space limitations though the $Zipf$ distribution yielded very similar trends. Each experiment had 5 seeds to explore various matrix configurations. The results presented are the average of the 5 independent runs with different seeds. Furthermore, the times presented are the time consumed to factorize the matrix using one iteration calculated by running each factorization for 15 iterations then dividing the total time by 15, unless stated otherwise . The module generates matrices with a specified sparsity $Gaussian$ or $Zipf$ distribution and a dimension $R^{m \times n}$ unless stated otherwise. We were keen to have the dimensions, m and n, as multiples of 2 to avoid as much as possible unbalanced matrix divisions.

4.1 Experimental Results and Analysis

The set of the experiments we chose are meant to reveal the performance of the different schemes we developed under different situations. We carried out experiments regarding sparsity. Another set of experiments present our block-based DNMF in comparison with our modified, CasDNMF. We generated matrices having 2^{23} cells to factorize. Table 1 illustrates the general parameters of the experiments run. We chose the parameters as near as possible to the problem that we are trying to solve which is the Web.

Table 1: General Simulation Parameters

Parameters	Values
Matrix Size	$2^{12}.2^{11}$
Factorisation Dimension	2^6=64
Sparsity	2^{-7}
Matrix Division Method	Block, Column, Cascaded
Speculation	Disabled
Dfs.block.size	256 mb

4.1.1 Matrix Sparsity

This experiment investigates the effect of changing matrix sparsity on the factorization time. Figure 2 shows that CasDNMF can perform up to 19% faster than the block-based when tested with a dense matrix. This can be attributed to the absence of the communication among the nodes in the first stage. This strategy results in huge savings when it comes to carrying out a matrix factorization. In addition to the parallelism that CasDNMF has in the first cascaded layer, the second layer exploits the parallelism to factorize the consolidated weight matrix W, gathered in lines 17- 18 and illustrated in Algorithm 2. Though the approach of

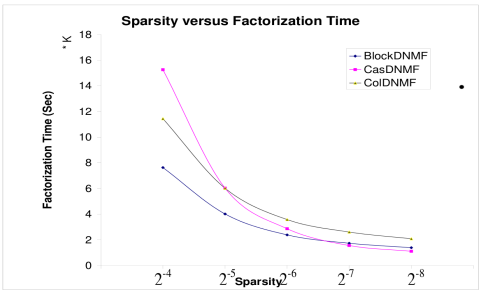


Figure 2: Matrix sparsity versus simulation time.

cascading led to good savings in the first layer, further parallelization techniques are required in order to reach higher levels of scalability for the second layer, which we achieved by using the BlockDNMF in order to factorize the second layer. In addition, Block-based DNMF results in lower time due to dividing the matrices on the nodes. CasDNMF provided faster factorization time by 17% compared with Block DNMF for the first matrix fed with sparsity of 2^{-4}. On the other hand running BlockDNMF matrices that are more sparse yielded faster factorization by an average of 24% compared with CasDNMF. Furthermore, block-based DNMF achieved faster factorization than the column based DNMF by 18% on average.

The sparsity range varies from 2^{-8} to 2^{-4}. The three schemes are almost following a linear relationship. By increasing the number of non zero cells in the matrix more processing time is required to calculate the matrix factorization. On the other hand, it can be observed from the same graph that CasDNMF provided faster factorization for the same matrices used on the block-based DNMF. This can be attributed to the amount of parallelism that is in the cascading factorization. In the cascaded factorization after the matrix is divided on the stations there is no communication among the stations until the factorization is complete. On the other hand, the second cascaded phase requires the factorization of a fat matrix which acts as a bottleneck. As a result, the average time consumed to factorize the matrix using the cascaded method increased.

4.1.2 Dimensionality

We aim to analyze the effect of changing the factorization dimensionality, k on factorization time. Figure 3 shows the results of the 3 factorization schemes. The graph illustrates that block-based distributed NMF provides faster factorization time by 15% on average. We can conclude that the time spent to factorize a matrix is directly proportional to the dimensionality chosen over the range of values illustrated. This can be attributed to the size of the matrices in the memory. Multiplying matrices where one of them is with lower dimensionality results in faster factorization. The graph shows the time increasing as the factorization dimension increases. This can be attributed to the need to process more data. Further, the number of the reduced records decreases as the number of zero records increases. The three

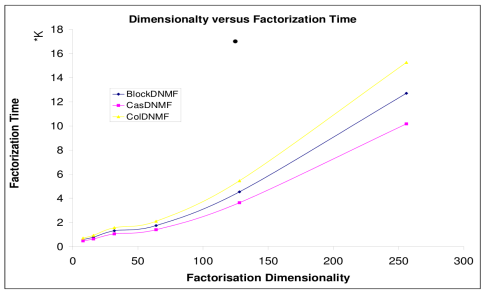


Figure 3: Dimensionality reduction versus factorisation time.

schemes exhibit very similar activity towards the fed matrices. In addition, more intermediate results are produced when the number of non zero cells increases.

4.1.3 Convergence Error

This experiment illustrates the change in convergence values as the number of iterations increases. Figure 4 shows the convergence error for both DNMF and CasDNMF. Both BlockDNMF and ColDNMF have identical convergence values. Block DNMF converges to higher degrees than CasDNMF with an average of 23% for the same number of iterations. This can be attributed to the way CasDNMF is designed since there is no communication among the cluster nodes during the factorization in the first phase. As a

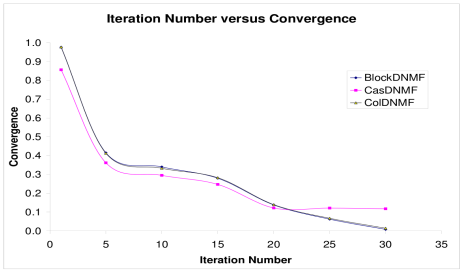


Figure 4: Number of iterations versus convergence value.

result, W and H are computed based on fewer number of cells. This leads to affecting the level of convergence. Equation 5 was used to calculate the convergence of the schemes developed.

$$Convergence = \frac{\sum_{n=1}^{n=num}((|O_n - C_n|)/C_n)}{num} \quad (5)$$

O_n: non zero n^{th} element in the V matrix.
C_n: non zero n^{th} element in the $V^{computed} = W \times H$.
num: total number of non zero cells in matrix V.

The final convergence value is 0.012% which was identical for both column based and block-based NMF. We developed the formula illustrated in equation 5 in order to calculate the convergence of the resulting factorization. Cascaded DNMF provides a very similar trend yet it converges at a lower rate than DNMF, with average convergence equal to 2.6% per iteration approximately.

Not only CasDNMF converges at a slower rate but also it reached a plateau starting the 20^{th} iteration. This can be attributed to the way the iterations are done in cascaded NMF. On the other hand, increasing the number of iteration for CasDNMF results in reaching an error of 0.8% in the 40^{th} iteration. As a result, CasDNMF needed more iterations in order to reach similar convergence values compared to BlockDNMF. In the first phase the updating rules are limited to use only the cells on the station. As a result, to update a cell only part of the matrices V, W and H are used to calculate its value. Consequently, the value gets stuck in a plateau.

4.1.4 Scalability

Figure 5 illustrates the time required for the factorization when the number of nodes is increased. Increasing the resources shows that blockDNMF performs much better than both CasDNMF and colDNMF by a factor of 23%. This shows that blockDNMF has higher level of parallelism. Increasing the resources for CasDNMF does not lead to much gains in the speedup. This can be attributed to the increase in the intermediate matrices after the first cascading phase. The inflation in data at the intermediate stage leads to the need of more communication in order to compute the factorization in the second cascading phase. The data inflation re-

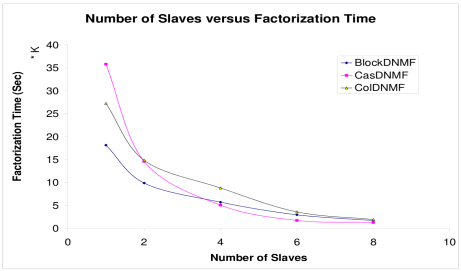


Figure 5: Number of nodes versus factorization time.

sults in restricting the amount of gains that can be achieved by using the cascaded scheme. This can be attributed to the amount of data that needs to be exchanged among the nodes. The more the number of nodes the more the data exchanged; as a result, it dominates the overall computational time.

4.1.5 Dimensionality with Zipf Distribution

We analyze the effect of drawing numbers from a *Zipf* distribution on the time needed to perform a factorization. We fed the module with different matrices each having different dimensionality factorization requirement, where the values are drawn from a *Zipf* distribution. Power distributions arise in several situations one of them is in social networks. The participation of users interaction through social networks follow a *Zipf* distribution [7]. The importance of this finding stems from detecting active users, influential information spreaders and topic initiators. As a result, we carried out experiments where the data is drawn from a *Zipf* distribution as stated before. Figure 6 shows the processing time required for dimensionality reduction. The trend is very similar to the one obtained for values drawn from the *Gaussian* distribution set. We observed that the standard

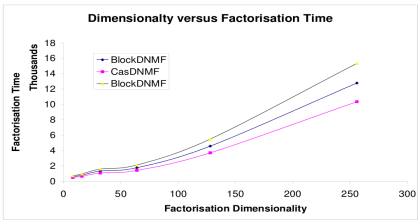


Figure 6: Dimensionality reduction versus factorization time.

deviation for the computational time of the *Zipf* distribution is about 9% more than the *Gaussian*, over the range of values measured.

5. CONCLUSION & FURTHER RESEARCH

Through our research we developed a valuable tool that can be used to reduce dimensionality of data, perform semantic indexing and prediction. Further, it can be used to provide clustering. We developed a scalable approach through which matrices that might not fit into memory could be factorized as well as matrices that fit into memory. Block-based DNMF provided better performance on less sparse matrices which is usually occurs with Web data. On the other hand, cascaded DNMF converges slower which requires more iterations in order to reach the same level of convergence as block-based DNMF. We plan to run further test to fully asses the performance of the methods developed under different parameter combinations. We are planning to use the scheme developed in order to provide diversified recommended resources and predict missing data. We plan to use the current techniques in disaster management and recovery for discovering latent relations among events. In addition, we are currently developing techniques to discover discussions and emerging events using the tools developed for safety decisions and disaster management.

We focused on scaling the techniques developed. Further experiments for latent prediction of the modules are crucial in order to explore their ability to provide prediction and clustering. Another venue for future research is dealing with updates. Incorporating different types of updates is an interesting challenge that needs to be taken into consideration in order to build an efficient way to calculate the matrix factorization. Applying the techniques developed on other data stets can be interesting to reveal other research caveats.

6. ACKNOWLEDGEMENTS

This work has been supported in part by Amazon grant. We are thankful to the Amazon cloud computing team for their help and advice on porting the algorithm for running on the Amazon cloud. The work has also been supported by IMT Lucca Italy.

References

[1] M. Bhandarkar. Mapreduce programming with apache hadoop. In *Parallel Distributed Processing (IPDPS), 2010 IEEE International Symposium on*, page 1, april 2010.

[2] R. Bradford. An empirical study of required dimensionality for large-scale latent semantic indexing applications. In *Proceeding of the 17th ACM conference on Information and knowledge management*, pages 153–162. ACM, 2008.

[3] D. Carstoiu, A. Cernian, and A. Olteanu. Hadoop hbase-0.20.2 performance evaluation. In *New Trends in Information Science and Service Science (NISS), 2010 4th International Conference on*, pages 84 –87, may 2010.

[4] A. Cichocki and R. Zdunek. Multilayer nonnegative matrix factorization using projected gradient approaches. *International Journal of Neural Systems*, 17(6):431–446, 2007.

[5] J. Dittrich, J.-A. Quiané-Ruiz, A. Jindal, Y. Kargin, V. Setty, and J. Schad. Hadoop++: making a yellow elephant run like a cheetah (without it even noticing). *Proc. VLDB Endow.*, 3:515–529, September 2010.

[6] C. Dong, H. Zhao, and W. Wang. Parallel nonnegative matrix factorization algorithm on the distributed memory platform. *International Journal of Parallel Programming*, 38(2):117–137, 2010.

[7] P. Ekler and T. Lukovszki. The accuracy of power law based similarity model in phonebook-centric social networks. In *Wireless and Mobile Communications (ICWMC), 2010 6th International Conference on*, pages 209 –214, sept. 2010.

[8] P. Hoyer. Non-negative matrix factorization with sparseness constraints. *The Journal of Machine Learning Research*, 5:1457–1469, 2004.

[9] D. Lee and H. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[10] D. Lee and H. Seung. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13, 2001.

[11] H. Lin, X. Ma, J. Archuleta, W.-c. Feng, M. Gardner, and Z. Zhang. Moon: Mapreduce on opportunistic environments. In *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, HPDC '10, pages 95–106, New York, NY, USA, 2010. ACM.

[12] C. Liu, H. Yang, J. Fan, L. He, and Y. Wang. Distributed nonnegative matrix factorization for web-scale dyadic data analysis on mapreduce. In *Proceedings of the 19th international conference on World wide web*, pages 681–690. ACM, 2010.

[13] X. Pei, T. Wu, and W. Yan. The approach based on dividing and conquering for non-negative matrix factorization. In *Intelligent Systems and Applications, 2009. ISA 2009. International Workshop on*, pages 1–3. IEEE.

[14] R. Peter, G. Shivapratap, G. Divya, and K. Soman. Evaluation of svd and nmf methods for latent semantic analysis. *International Journal of Recent Trends in Engineering*, 1(3), 2009.

[15] S. Seo, E. Yoon, J. Kim, S. Jin, J.-S. Kim, and S. Maeng. Hama: An efficient matrix computation with the mapreduce framework. In *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*, pages 721 –726, 30 2010-dec. 3 2010.

[16] F. Shahnaz, M. Berry, V. Pauca, and R. Plemmons. Document clustering using nonnegative matrix factorization. *Information Processing & Management*, 42(2):373–386, 2006.

[17] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273. ACM, 2003.

[18] Z. Zhang and X. Zhang. Two improvements of nmf used for tumor clustering. In *1st Int. Symposium on Optimization and Systems Biology*, pages 242–249, 2007.

Filtering search results using Explicit Feedback

Varun Gupta[1], Neeraj Garg[2], Kapil Jhamb[3] & Lakshya Bhagat[4]
[1,2]Computer Science &Engg., MAIT, GGSIPU, India, [1]varunguptacs@gmail,[2]neeraj_garg20032003@yahoo.co.in
[3]Department of information systems, George Mason University, VA, USA, kjhamb@masonlive.gmu.edu
[4]Department of Computer Science, Columbia University, NY, USA, lb2787@columbia.edu

Abstract--*Most web search engines consider only the search query for searching. Semantic and usage ambiguity of the search keywords results in noisy search results and many of them do not match the user's search goals. This paper presents the design of an interactive and Search Bot, which operates as a filtering agent for a user by simulating the user activity of finding the relevant search results. It learns from experience and improves its performance over time. It focuses on obtaining user's requirement or search intention from the search query, and then delivering results accordingly. Firstly, the user trains the system for to his search intention by either doing binary classification of the search results or by giving them a relevance rating on a scale of 1 to n. Training is followed by knowledge representation and inference, and then by reasoning and analysis of the new search results to determine their relevance classification. The technique is based on probabilistic reasoning. It finds application in information retrieval from databases, news searching and detection of spam mails.*

Keywords-Information Retrieval; Web search; search intention; explicit relevance feedback; user training

1. Introduction

Many web search engines use only keywords for analysis in searching. They don't encompass the users search requirement in their searching process. The searched keywords in the search query might have several meanings and usages, and thus many unwanted results are expected in the list of search results returned by the search engine. Even if the search results listed correspond to the correct interpretation of the query, still many unwanted and irrelevant results are present. Moreover, different users may expect different search results from a given search query. Thus there is a need of a user-centric searching process, which filters results according to user-specific search goals. This is achieved by taking feedback from the user to know his preferences.

When the search results are displayed for a search query, the user can train the system as per his requirements. He can choose amongst two training methods. In the first one, the user is supposed to do binary classification i.e. to simply mark the search results as relevant or irrelevant. The second method requires the user to rate the results' relevance on a scale of 1 to n. By inferring knowledge from this input, the technique then filters the search results in 3 cases:

1. Training using a search query and filtering results for another query in the same search domain.
2. Training using a search query and retyping the search query later to get filtered results.
3. Training using the first few results of a search query, and filtering the rest of the results in a one-time search.

These are achieved by creating a training set for each search query and for a search domain. When the search intentions for different queries can be logically categorized into one logical category, they are said to lie in a common search domain, e.g. the search query "IR" which has multiple inferences like Information Retrieval, Indian Railways, International Rectifier. Say, the system is trained for search intention pertaining to Information Retrieval. Now if search query "IE" is searched and the user is looking for results pertaining to Information Extraction, which is logically related to Information Retrieval, not for Internet Explorer, then the search intention for the 2 search queries can be said to occur in a same search domain.

An intelligent bot is a program that operates as an agent for a user or another program or simulates a human activity based on some knowledge which is either stored already or gained from experience [17]. Since the system based on the proposed technique learns from the training given by the user, and then performs reasoning and filtering on the basis of the knowledge acquired by inferring mechanisms, it is an intelligent search bot. The system working on this technique learns from experience and can be trained by the user even after he gets to see the filtered results, which can further enhance the precision of the filtering process.

The rest of the paper is organized as follows. Section II mentions the related work and Section III discusses the proposed technique in detail. Section IV discusses the implementation of the technique, while Section V mentions the experiments and evaluation. Finally, the conclusion and future work are given in Section VI.

2. Related Work

User's search goals can be communicated to the system in form of implicit and explicit feedback. Implicit feedback comprises a collection of user behaviors and activities like his page visits, searches, his routes from one site/page to another etc [4]. A variety of methods have been developed to interpret implicit user feedback and to use them for personalized search. [5] presents a new approach of

developing ontology based user profiles for web searching. It models the user's search intentions by the process of PTM (Pattern-Taxonomy Model). Wu et. al. [12] also used a pattern-based taxonomy rather than single words to represent documents for modeling user profiles. R.Y. Shtykh et al. [4] have mentioned layered user profiling to capture user search intentions derived from implicit and explicit feedback, where an interest change driven profiling mechanism was used for document evaluation and profile generation. Since the user profile information is stored locally on user's machine, portability of the user profiles seems to be a major issue such systems. Takehiro Yamamoto1 et al. [6] discussed an editable or a personalized browser which models user's intentions by tracking changes made in the search queries to re-ranks the web results. They also developed a query expansion mechanism, wherein a system sends expanded search queries to search engines, according to the past activities of the user [7]. Kinam Park *et al.* used users' web search logs to identify their search intentions [9]. Here he mined user's search intention from the query and represented it in an intention map. In the 3 systems proposed above, portability of the solution seems to be a problem because web search logs and the user activities are stored locally on his system. The implicit feedback is very advantageous as it doesn't interrupt the user's activities, but it is not very accurate because it is just a system's judgment of the user's intentions, and it may not accurately reflect the user's search intention in each case.

We thus explore the domain of explicit feedback, which is more accurate and credible as the user explicitly gives his preferences to the system. Bharat K. [10] developed a "Search Pad" which maintained a log of search contexts and demonstrated very good results. The "Feast" technique makes use of micro indices by the user as explicit feedback presets a "micro search" mechanism [8]. Another example of explicit feedback being used is when the user marks an email as spam, and then mail client filters off similar emails. A technique which models users search intention by taking user feedback in form of binary classification of search results has been discussed in [11]. It uses decision trees for modeling user search intention. The major drawback in this approach was that the search results could be only classified as either relevant of irrelevant by the user whereas many search results are not completely relevant or irrelevant. This paper presents an alternative and a better approach to model users search intention by firstly allowing him to give a relevance rating to the search results, and secondly, by using probabilistic reasoning for extracting knowledge from the training set provided by the user. It overcomes the inaccuracies involved in implicit feedback modeling techniques, and filters the search results not only according to the search context as in "Search Pad", but also according to the words and data in the new search results. Although taking explicit feedback accounts for training overhead, the higher precision of the filtered results makes up for efforts and time lost in training the system, by saving greater time and effort in most cases. There is also a provision of creating domain specific profiles, where training is not required for each query belonging to a domain for which the system has been trained in the past. This technique can be used be complemented with an implicit feedback technique for even more accurate results and lesser training overhead.

3. Proposed Technique

A technique based on machine learning and probabilistic reasoning has been formulated. It involves 3 major stages: user training in form of his input, knowledge inference, and reasoning with new search results to determine their relevance classification to filter the irrelevant ones. The user first provides training input which can be done by two different means. He can either classify the search results as simply relevant or irrelevant, or rate the search results on a scale of 1 to n. This input is represented in a structured manner, and then knowledge is inferred from the facts obtained from the user. This knowledge is used to reason with the new search results to determine their relevance. The steps used are explained in the following paragraphs.

Step 1. User Training

Here the user conveys his requirements or search intention for the search query to the system. Since the searching has to be made user-centric, accepting user preferences is very crucial in the process. The user has to perform binary classification or to give a relevance rating on a scale of 1 to *n*, with 1 being the least relevant and *n* being a positive integer greater than 1. For instance, when n=5, the user can rate each result by assigning it any integral value from 1 to 5. The ratings could be displayed to the user in form of relevance degree as:

1- Completely irrelevant; 2- Somewhat Irrelevant; 3-Can't Say; 4- Somewhat relevant; 5- Completely relevant

Similarly n can take any value greater than 1 depending on the user's requirement and the relevance degrees can be displayed to the user accordingly. It is recommended that $2 \leq n \leq 5$, as a higher value of n makes it difficult for the user to assign an accurate rating to each search result. When the user does binary classification of the search results i.e. classifies the search results as relevant or irrelevant, then it is interpreted by the system as a rated input where n=2. So it can be said that in binary classification, "Irrelevant" is interpreted by the system as Rating = 1 and "Relevant" as Rating = 2.

When user classifies or rates a search result, it is termed as a training instance, and the set of search results he rates is called a set of training instances, referred to as "S". This training set data is stored in 2 training sets: Ss_{SQ} i.e. training set for a search query SQ, and Ss_{SD} $i.e$ training set for a search domain SD.

Step 2. Knowledge Extraction

The training input provided by the user is in form of raw data which need to be processed, cleaned and structurally represented in the system so that knowledge can be inferred or extracted from it. The knowledge extraction process follows a certain set of steps.

Step 2.1. Gathering Information from the Training Data

An exhaustive list of all the words that occur in all the results in the training set, along with some counts is prepared. For each word w in the list, following counts are calculated:

1.$N(RC = k)_w$: No. of search results with Rating = k, which contain a word w. This value is computed for all values of k=1,2,3…n.

2. $N(Total)_w$: Total no. of search results containing word W_i.

3. P(R| W_i)= Probability of the search result being relevant if word W_i occurs in it. Its formula is discussed later.

It is important to note that the number of occurrences of a word w in a search result with Rating Category=*k* is not counted, but the number of search results with Rating Category=*k*, containing the word w is counted.

In the Step 1 (User training), each search result is given a rating category, which is interpreted by the user as "Completely relevant", "Somewhat relevant" etc. These ratings can be interpreted by the system in terms of probability i.e. Probability of a search result being relevant if it is given a Rating Category=*k*. It is calculated as:

$$P_j(Search\_Result = Relevant|Rating\_Category = k)$$

or
$$P(R|RC=k) = (k-1)/(n-1) \quad (1)$$

where *k* = Rating of the search result.

n= No. of integers used to rate the search results.

The *P(R|RC=k)* value is formulated such that for a search result with lowest rating its value it equals 0, while for the maximum rating value, it equals 1.

Table 1. $P_j(R|Rc=K)$ Values For K= 1 To 5

Rating Category	$\boldsymbol{P_j(R|RC=k) = (k-1)/(n-1)}$
1	1-1/ 5-1 = 0
2	2-1/ 5-1 = 0.25
3	3-1/ 5-1 = 0.5
4	4-1/ 5-1 = 0.75
5	5-1/ 5-1 = 1

P(R| W_i) i.e. the probability of a search result being relevant given the word W_i exists in it, is calculated as

$$P(R|W_i) = \frac{\sum_{k=1}^{N}(N(RC=k)_{W_i} \times P_j(R|RC=k)}{N(Total)_{W_i}} \quad (2)$$

Different values of *P(R| W_i)* are shown in Table 2. Considering the word "Metal", its\ *P(R| W_i)* is :

$$P(R|Metal) = \frac{(0\times0)+(1\times.25)+(3\times.5)+(5\times.75)+(16\times1)}{25} = 0.86$$

Considering the case of n=2, where the user trains by doing binary classification of the results. Say, a word W_k occurring in 6 relevant results, and 3 irrelevant results, then

$$P(R|W_k) = \frac{(0\times3)+(7\times1)}{10} = 0.7$$

In this case the formula seems to be more intuitive i.e.

$$P(R|W_k) = \frac{No.of\ Relevant\ results\ with\ this\ word}{No.of\ total\ results\ with\ this\ word}$$

Table 2. Sample Wordlist For N=5 & Query "Palladium"

Word	$N_{RC=1}$	$N_{RC=2}$	$N_{RC=3}$	$N_{RC=4}$	$N_{RC=5}$	N_{Total}	**P(R|W_i)**
Metal	0	1	3	5	16	25	0.86
Platinum	1	1	0	6	11	19	0.82
The	31	23	40	32	41	167	0.54
Perform	10	2	1	0	0	13	0.77
Of	13	18	24	19	29	103	0.58
Mining	0	0	0	0	14	14	1.0
Ounce	3	0	0	4	8	15	0.73
&	1	1	1	0	2	5	0.66
London	5	2	1	0	1	11	0.18
After	3	1	1	0	2	8	0.34
Dance	7	0	0	0	0	7	0.0

Step 2.2. Data Cleaning

The word list obtained above has some inappropriate data which need to be cleaned. It contains parts of speech and special characters which, in most cases, are not required for analysis as they do not truly reflect the relevance of a search result. If this data is used for analysis in the later steps, it may give inconsistent and inaccurate results. Thus such data act as noise and need to be removed from the training data. This step has 2 stages.

Step2.2.A. Removing parts of speech and special characters- This stage removes some parts of speech (e.g. articles, prepositions, conjunctions, pronouns) like: to, it, the, a, am, from etc., and special characters like '.','?', '&', '%' etc. This step may not be required in all the cases, for e.g. when the search query has the part of speech or a special character, thus this step may be modified and is optional.

Step2.2.B: *Adjusting extreme probabilities*- In this step, we adjust the extreme values of *P(R|W_i)*: 1 and 0, because these numbers are potentially over-fitting. This is because it wouldn't be correct to assume the *P(R|W_i)* for a word to be 100% or 0% since there might be a search result contradicting these extreme values. Also, in calculation of value of P(R) using (5), if there are attributes with $P(A_i)$ with values 0 and 1, then P(R) value will attain $0/0$ form. So we adjust the *P(R|W_i)* value as

I*f P(R|W_i) = 1* , then adjust *P(R|W_i) to 0.99.*

If P(R|W_i)= 0 , then adjust *P(R|W_i)to 0.01.*

The words obtained after the Data Cleaning stage are referred to as Attributes and their corresponding $P(R|W_i)$ are referred to as $P(R|A_i)$.Thus a new list called "Attribute List" is prepared which consists of all the "Attributes," their corresponding $P(R|A_i)$ and $N_{RC=k}$ values.

Table 3. Sample Word List For The Word List In Table 2

Attribute	$N_{RC=1}$	$N_{RC=2}$	$N_{RC=3}$	$N_{RC=4}$	$N_{RC=5}$	N_{Total}	$P(R\|A_i)$
Platinum	0	1	3	5	16	25	0.86
Metal	1	1	0	6	11	19	0.82
Perform	10	2	1	0	0	13	0.77
Mining	0	0	0	0	14	14	0.99
Ounce	3	0	0	4	8	15	0.73
London	5	2	1	0	1	11	0.18
Dance	7	0	0	0	0	7	0.01

Step 3. Reasoning&Filtering new search results

The technique is now supposed to determine the relevance of the new search results. It is divided into 2 steps.

Each Attribute A_i is matched with the words in the new search result to check whether the Attribute exists in the new search result or not. If an Attribute is found in the search result, is referred as an "Attribute hit", otherwise an "Attribute miss". For each *Attribute hit*, we have probability of that particular search result being relevant, represented as $P(Search\ Result=Relevant|A_i)$ or $P(R|A_i)$, and its values are shown in Table 2.

Considering only one *Attribute hit*, however, is not sufficient as each search result might have multiple *hits*. All the *Attribute hits* in the search result, and their corresponding $P(R|A_i)$ values are used for calculating the probability of a new search result being relevant. The Bayes' theorem for combining independent probabilities is used:

$$P(R) = \frac{\prod_{i=1}^{n} P(R|A_i)}{\prod_{i=1}^{n} P(R|A_i) + \prod_{i=1}^{n}(1 - P(R|A_i)} \qquad [18] \qquad (3)$$

where, $P(R)$ = Probability of the new search result being relevant.

N = Total no. of Attribute hits with search result

$P(R|A_i)$ = Probability of the search result being relevant given that an Attribute A_i hit.

This formula assumes the probabilities $P(R|A_i)$ are independent.

For instance, a new search result has following *Attribute hits*: "mining", "gas", "perform" and "London" .Say $P(R \mid Mining)$ = 0.95 and $P(R \mid gas)$ = 0.0.87, $P(R \mid perform)$=0.07 and $P(R \mid London)$=0.35 then

$$P(R) = \frac{0.95 \times 0.87 \times 0.07 \times 0.35}{(0.95 \times 0.87 \times 0.07 \times 0.35) + (0.05 \times 0.13 \times 0.930.65)} = 0.8381$$

This shows that if the above mentioned 4 words occur in the new search result, then the probability of that search result being relevant is 0.8381**.**

Now we have calculated the probability of relevance of the new search results. This probability can be used to display the search results to the user in different forms like:

1. *Filtering off irrelevant results*- The value of P(R) alone is not sufficient enough to determine the relevance of the new search result. There must be a threshold value with which P(R) can be compared. This specific value can be termed as "Filter Threshold value (FTV)" for this technique and is used as:

If $P(R) < FTV$, the result can be classified as irrelevant.

If $P(R) \geq FTV$, the result can be classified as relevant.

Logically, FTV value should be *0.5*. But this value can be adjusted to a higher or a lower value according to the level of relevance quotient required. For instance, if FTV =0.7, then only the new search results with $FTV \geq 0.7$ will be classified as relevant by the technique, and rest all the irrelevant results will be filtered out and won't be displayed to the user.

2. *Displaying the search results in decreasing order of P(R)*- This shows the most relevant results on the top while the least relevant ones at the bottom. Since it doesn't remove the results with low values of P(R), it makes sure that no result is missed out even if the system wrongly judges the P(R) values.

3. *Categorizing results as per ratings*- The P(R) values can be converted to ratings and the results can be displayed in descending order of ratings. Table 4 shows this conversion can be done and Figure 5.2 shows an example of implementation of this sorting. Say, the search results were rated from a scale of 1 to 5 during training, then the new search results can be rated on a scale of 1to 5 on the bases of their $P(R)$ values as:

Table 4. Ratings According To Values Of P(R)

P(R)	Rating	Displayed to the user as
0.0-0.20	1	Most Irrelevant
0.21-0.40	2	Somewhat Irrelevant
0.41-0.60	3	Can't Say
0.61-0.80	4	Somewhat Relevant
0.81-1.0	5	Most Relevant

If the new results are to be rated on a scale of 1 to *n*, then the cut-off values for $P(R)$ can be set to k/n where *k* is the rating which ranges from 1 to *n*. When *n=2*, i.e. when user chose to do binary classification of the search results, then the cut-off value would be *1/2* or *0.5*.

Thus, this technique accepts the user's search requirements from the search query through ratings or binary classification, stores and cleans this data, represents these facts in form of an attribute list, analyzes the new search results of the same query or a new query to find their

probability or degree of relevance and then allows different ways to display the search results to the user.

4. Implemntation

The Search bot is a middleware between the user and a search engine. The search bot fetches the search results from an existing search engine, filters and analyzes all the results, and displays the relevant ones to the user. So the search results for a search query can be fetched from a search engine like Google, Yahoo, Baidu etc. using Application Programming Interface (API). The search bot was developed in a .net framework using Microsoft Visual Studio 2010. To handle the storage of data for the search bot, a database or a data store is used. The user profiles, training data, knowledge representation and inference structures are stored using SQL server 2008. Active Data Objects (ADO.net) was used for interfacing the application with the database.

As explained in Section 1, different search domain profiles are created for each user to filter the results according to their own preferences. Say a computer science researcher could have different search domains as different areas of research like Information Retrieval, Networks, Parallel computing etc. So if the Retrieval" profile will show results corresponding to the abbreviation of IR as Information Retrieval, while Networks Profile will show results corresponding to abbreviation of IR as Infra Red.

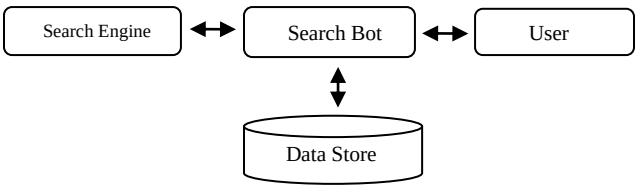


Fig.1. Block Diagram of the searching system using this technique.

In this technique, when the new search results are displayed to the user after being sorted, he can still rate all or some of them, and this training set can be added to the existing training set for the search query or search domain profile. All the knowledge extraction steps will be applied again on this combined training set, and a new attribute list will be prepared with having new values of $P(R|A_i)$. This new and improved attribute list will be referred to, if the same or a new search query is searched and it will result in more accurate filtering. This technique, thus, can learn even while it is being used, and can enhance its precision with time.

For web searching, the user can use the technique in 3 ways:

1. Training using a search query and filtering results for another query in the same domain profile.
2. Training using a search query and retyping the search query later to get filtered results.
3. Training using the first few results of a search query, and filtering the rest of the results in a one-time search.

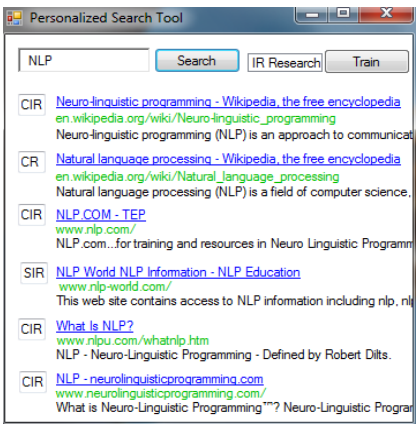
In case of information retrieval from a data store or a database, the existing searching system can continue using its own searching mechanism to produce the search results which are fed into the new filtering system, which analyses and filters the search results.

In case of detection of spam mails, the user flags the spam mails like the way he did binary classification of the search results in case of web searching. But here different domain profiles are not required as there are no search queries in this case. Thus one comprehensive training set is prepared, and each time the user marks the relevance classification of the emails, the training input is added to the comprehensive training set, through which updated word list and attribute list are prepared. When new mails arrive in the inbox, their relevance is automatically determined by applying the reasoning process (Step 3) using the updated lists and tables, and spam mails are detected and filtered out.

5. Experiments

A set of experiments were conducted in order to evaluate the precision and performance of the proposed technique. A search tool using this technique was developed as per the design instructions given in Section IV. Participants included students, business analysts, researchers, businessmen etc., each of whom was asked to use their own search queries for the experiment. A total of 1000 search queries were used in the experiment. The search queries could be a simple search keyword or a complex multiple word query. The search intentions of each participant for his search queries were recorded in advance in form of a brief description about the participant's requirement from the search query. Firstly, the search queries were searched with Google search engine, and the relevance for the first 20 results for each search query was judged by the users. For the first 10 results, the users marked a relevance rating or classified the results as relevant or irrelevant, then pressed the 'Train' button and then again judged the relevance of the next 20 unseen search results. The search queries for which the search results were expected to change with time, the users were asked to check the first 20 search results for their search query after one month. For others, the users adjudged the relevance of the first 20 results after the 10^{th} result. More importantly, the users were also asked to test the relevance of the search results for a new search query in the same domain as the search query which was used for training.

All the relevance judgments from the users in the experiment were recorded to compare the precision, accuracy, MAP etc. before and after the use of the tool using the proposed technique. Precision is the fraction of retrieved results that are relevant, whereas Accuracy is the fraction of the classification done by the filtering system which are correct and Recall is the fraction is the fraction of relevant



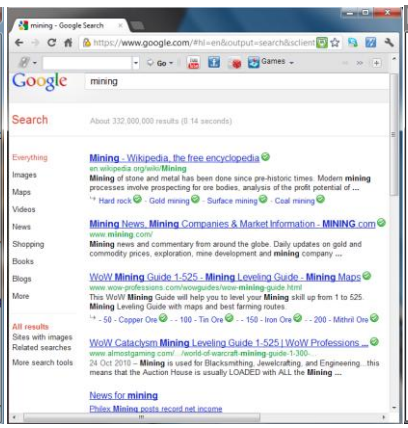


Fig.5.1Search & Training Window For "NLP" Fig.5.2 Google Search For "Mining" Fig.5.3 Sorted Results For "Mining"

documents that are retrieved by the system. Also, all the events in the experiment were timed for each user to determine the time saved or lost with the use of new technique. The following observations were made-

Observation 1

Here the search query used to train the system was "NLP". In Fig. 5.1, the user trained the system by marking the relevance rating for the results for a search query as CR, CS, SIR etc. for Completely Relevant, Can't Say, Somewhat Irrelevant respectively. Then he mentioned the name of the search domain profile as "IR Research" and he pressed the "Train" button. Then he searched for another query "Mining" using "IR Research" profile. His intention was to search for mining of data e.g. Text Mining, Web mining etc and not for mining of stones and metals. Figure 5.2 shows the search results displayed by the Google search engine for the query "Mining" and Fig. 5.3 shows the sorted results using the technique. Here the system showed the results corresponding to Mining of data at the top, while those corresponding to mining of stones, metals at the bottom. The user can still mark the relevance ratings of the sorted results and press the "Train" button to further enhance the precision of the searching process.

Observation 2

Figure 6 shows the comparison of precision of Google search engine and the tool using the proposed technique. The X-axis
depicts the search query number, which are numbered from Q1 to Q1000 and Y-axis depicts the precision values in percentages. Since regular search engines don't rate or categorize the results in relevance degree, a comparison of precision of rated classification with the new technique can't be compared. So we compare the case of binary classification. A few examples of search queries with the search intention, used in this experiment are given in Table 4. The precision values were noted for 3 cases:

Case 1. Search queries are searched on Google Search Engine (Blue color). Average Precision = 68.34%

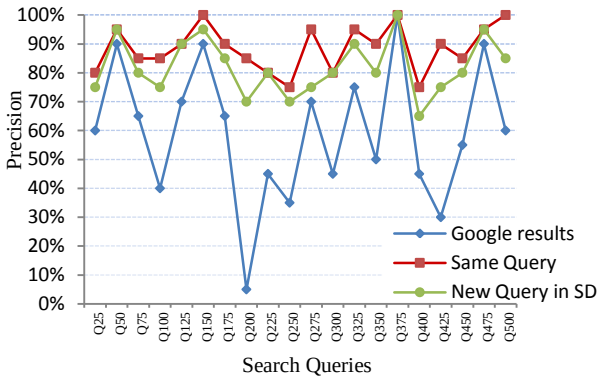


Figure 6. Search Precision for different search queries.

Table 4.Examples of Search Queries&their intention for Figure 6

Query no.	Query	Intention
Q3	Jaguar	Car co.
Q5	Versace	Apparel brand
Q8	TIME	Name of training institute
Q14	Sample papers for SAT exam	Sample Question papers
Q15	Grammy awards	Music awards
Q17	What is in your wallet	Tagline for which company?

Case 2. Same query is searched again after training. (Red color). Only the search results for which the system hasn't been trained are considered for precision calculation. Average precision= 94.82%.

Case 3. System is trained for a search query and a new search query in the same search domain profile is searched (Green color). Average precision = 87.18%.

It is evident from the experimental results in the graph that precision of the 2 cases using the technique is much higher than that of Case 1 where Google search engine was used. The precision value when averaged for all the queries

in Case 2 was greater than that of Case 3 because when the when the query used for training and filtering is the same, predictably the results' classification should be more accurate than in the case where different queries were used for training and filtering. It was observed that the precision of the Google search results varied a lot according to the search query and the user's search
intentions e.g. if the search query is "Apple", 19 out of the first 20 results are about the Apple IT co., and 1 result is about the apple fruit. So if the search intention is apple fruit, then precision would be 5%, whereas for Apple IT co. it would be 95%, but the precision of the proposed technique was higher in both the cases. For search queries with a single and an obvious interpretation like "David Beckham", "Samsung", "Macintosh" etc. the searching precision did not change much even after using the technique as Google returns very relevant results with extremely high precision and accuracy values for such queries. For complicated, long queries, or queries with multiple inferences, the technique significantly improved the precision.

Observation 3

Here we calculated the Normalized Discounted Cumulative Gain (NDCG) for the search results for all queries with and without the use of the proposed technique. DCG (Discounted Cumulative Gain) is a measure of effectiveness of a searching algorithm by calculating the usefulness, or *gain*, of a search result based on its position in the result list using a graded relevance scale of documents in a search engine result set. DCG till a particular rank position *p* is given by

$$DCG_P = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{log_2 i}$$

where rel_i = Graded relvance of the result at position i
NDCG normalizes the value of DCG for different queries, given as

$$DCG_P = DCG_P / IDCG_P$$

where $IDCG_P$is the Ideal DCG_P, in which the results displayed are in descending order of graded relevance when judged by the user. The comparison of NDCG values helps us in evaluating the proposed technique using graded relevance scales with regular search engines using binary classification. The value of NDCG varies between 0 and 1. Figure 7 shows the NDCG values for 500 queries for all the 3 cases mentioned in Observation 2. When this technique was used, the NDCG values were found to be much higher than with Google search engine, and again NDCG values in Case 2 were better than that of Case 3 due to the reasons for reasons cited earlier. NDCG value when averaged over all over search queries with Google search was .51 whereas for Case 2, it was .90 whereas for Case 3 was .847.

Observation 4

Figure 8 shows the graph representing the relation between the time saved and the no. of search results considered by the user, referred to as N_{SR}. No. of search results can be

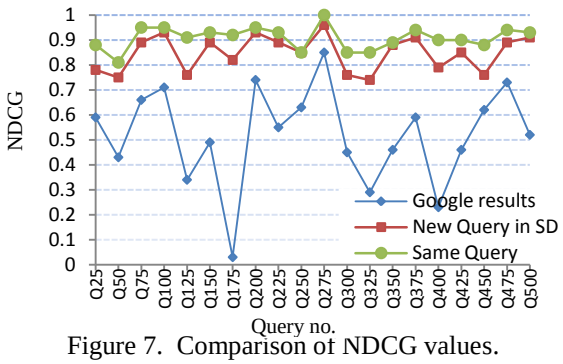


Figure 7. Comparison of NDCG values.

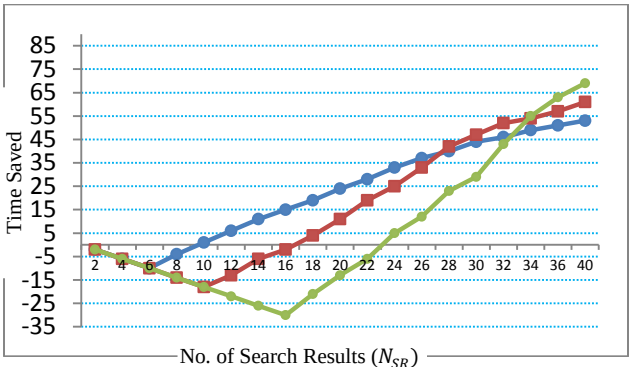


Figure 8. Time saved vs. No. of search results

divided into 2 categories: No. of search results used for training ($N_{Training}$) and the no. of search results which can be filtered by the tool after the training, referred to as$N_{Filtering}$. The graph shows that the time saved is negative for the first few search results. This is because the user has to put in extra time to mark the relevance classification of the search results he uses for training, which accounts for time lost or a negative time saved. For the next few no. of search results, when the training process is over, and the filtering starts, the time saved is increasing, but is still negative, because the tool is saving the user's time, but hasn't been able to cover up the time invested for training. Then the time gain becomes zero when the time invested in training becomes equal to the time gained due to filtering of irrelevant results. After that, time saved is positive and keeps on increasing, and the usage of the tool becomes very beneficial. This shows that the only investment of time happens for the search results used for training, and when the user starts getting filtered results for the same search query, or for a different query, the time saved becomes positive. The proposed technique is very beneficial in cases with greater $N_{Filtering}$, which always happens when we use an existing training set to filter the results for a new query. Even for cases where we search again for the same search query which was used to train, it was observed that for $N_{Filtering} > 3/4 N_{Training}$, time gained was positive for most cases. The different lines in the graph show that when $N_{Training}$value increases, the time lost

initially is more, but the time saved later on exceeds that for instances with lesser value of $N_{Training}$. Though a higher value of $N_{Training}$ accounts for greater time lost in training by either using more search results for training or by using search results of more queries, it saves time later on if larger no. of search results are explored by the user because the filtering precision of the tool increases with $N_{Training}$. So, the no. of search results or the no. of search queries used for training has to be selected in accordance to the no. of search results the user wants to explore by using the training set. When the search query is used only once such that there will be no other search queries in that profile, then the no. of search results for training can be any value from 5 to 10. If multiple queries will be searched in the same profile, then he can afford to train the system with more no. of search results, or with results of multiple search queries.

6. Conclusion and future work

This paper proposes a technique which can be used to sort the search results according to user's search intention which is conveyed to the system in form of explicit feedback. The system fetches the search results from a searching system, which performs searching on the bases of keywords in the search query. This technique then filters these results according to the user's requirements for the search query. Thus the search results which are displayed to the user are searched according to the keywords in the search query, and then filtered according to the user's requirements from the search query. The technique delivers reasonably high precision in filtering, saves time and can improve its precision with usage. There is no restriction on type or length of search queries. In fact the technique is more useful in multi-word or complicated queries because the searching precision for those with regular search engines is lower and the precision improves significantly by using the technique.

The feature of using search domain profiles which allows the training performed for a search query to be used for filtering results for a new query increases the utility of this technique. Since the results for a new search query can be filtered without any training requirement for that search query, this technique is better than the ones using explicit feedback for each query. Also the technique has been shown to find applications in information retrieval from databases and in spam mail detection.

However the user need to be careful in selecting the relevance ratings and in correctly grouping search queries into search domain profiles, since incorrect selection in these might lead to inaccurate filtering of the results. Since the search bot retrieves the results from a search engine and cannot change the searching process of the search engine, it can't change the recall value of the results, which is a limitation of this technique.

The proposed technique has been tested successfully with reasonably good results, but to improve the precision and NDCG further, we can combine explicit and implicit feedback mechanisms. It facilitates search intention based filtering even when the user is reluctant to train the system in case where there is no training performed in the past in the domain profile. This system of training with explicit feedback can be integrated with another system like the ones in [3,4, 5, 6, 9 and 10]. Taking feedback from multiple users for a given search query to produce filtered search results according to a common intention can also be worked upon.

Acknowledgment

We would like to show gratitude toward Dr. Ya'akov Gal, Division of Engineering and Applied Sciences, Harvard University for his valuable advice during the research.

References

[1] G.Salton and C. Buckley. Improving retrieval performance by relevance feedback. JASIS, 44(4):288-297, 1990.

[2] M. Iwayama. Relevance feedback with a small number of relevance judgements: incremental relevance feedback vs. document clustering. In SIGIR, pages 10–16, 2000.

[3] Roman Y.Shttkh and Qun Jin "Enhancing IR with User-Centric Integrated Approach of Interest Change Driven layered Profiling And User Contributions" in 21st IEEE International Conference of Advanced Information Networking And Applications Workshops (AINAW'07).

[4] J. J. Rocchio. Relevance feedback in information retrieval. In The SMART Retrieval System Experiments in Automatic Document Processing, pages 313–323, 1971.

[5] Xujuan Zhou, Sheng-Tang Wu, Yuefeng Li, Yue Xu, *Raymond Y.K. Lau, Peter D. Bruza, "Utilizing Search Intent in Topic Ontology-based User Profile for Web Mining", in proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006 Main Conference Proceedings)(WI'06)

[6] Takehiro Yamamoto1, Satoshi Nakamura1, and Katsumi Tanaka, "An Editable Browser for Reranking Web Search Results", in IEEE International Workshop on Databases for Next Generation Researchers, 2007. SWOD 2007.

[7] Zhengyu ZHU, Jingqiu XU, Xiang REN, Yunyan TIAN, Lipei LI, "Query Expansion Based on a Personalized Web Search Model", Third International Conference on Semantics, Knowledge and Grid, IEEE 2007.

[8] K.S. Kuppusamy, G. Aghila, ", FEAST - A Multistep, Feedback Centric, Freshness Oriented Search Engine", 2009 IEEE International Advance Computing Conference (IACC 2009) in Patiala, India, 6-7 March 2009

[9] Kinam Park, Taemin Lee, Soonyoung Jung, Heuiseok Lim, Sangyep Nam, "Extracting Search Intentions from Web Search Logs", in 2nd International Conference on Information Technology Convergence and Services (ITCS), 2010.

[10] Bharat K. "SearchPad: Explicit capture of search context to support web search" in Proceedings of the 9th International World Wide Web Conference, pp. 493-501, 2000.

[11] Varun Gupta, Tarun Gupta, Neeraj Garg, "Search Bot: Search Intention based Filtering based on Decision Tree based Technique", in proceedings of 3rd International Conference on Intelligent Systems, Modelling & Simulation(ISMS),IEEE 2012.

[12] Bayesian theorem for combining individual probabilities http://en.wikipedia.org/wiki/Bayesian_spam_filtering.

[13] E. Agichtein, E. Brill, S. Dumais, and R. Ragno, "Learning User Interaction Models for Predicting Web Search Result Preferences," in Proceedings of the ACM Conference on Research and Development on Information Retrieval (SIGIR), 2006, pp. 3-10.

[14] Bayesian theorem for combining individual probabilities http://en.wikipedia.org/wiki/Bayesian_spam_filtering.

Credo: A Framework for Semi-supervised Credibility Assessment for Social Networks

Ahmed Nagy
Carnegie Mellon University
NASA Mountain View
ahmed.nagy@gmail.com

Jeannie Stamberger
Carnegie Mellon University
NASA Mountain View
jeannie.stamberger@sv.cmu.edu

1. ABSTRACT

Social networks enable users to share information about events and themselves. We present a framework that assesses the credibility of messages of a social network. We offer a semi-supervised framework to judge the credibility of a message, *Credo*. The framework is based on PageRank. We test our model on data collected from Twitter. We find that our model outperforms the baseline, Bayesian networks. We designed several models that consider the poster-message, poster-reporter-message and poster-reporter-top-K-messages. *Credo* considers the trustworthiness of the poster, reporters and the credibility of the top K semantically similar messages. We were able to improve the perforamance of the proposed model by including features such as the semantic relatedness of messages and posting characteristics. Our model performs 18% better at ranking the credibility of a piece of information than baselines used.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Storage and Retrieval—*Knowledge Extraction*

Keywords

Knowledge Extraction, Data mining, Credibility, Twitter

2. INTRODUCTION

Rumor detection is an essential aspect in judging information usefulness. Data credibility is considered one of the important characteristics of useful data. The ability to spread rumors and spam has made micro blogging tools such as Twitter a target for spammers and rumor spreaders [12, 20]. We will start by defining the essential and the relevant terms to our research. Later we present the most important techniques that are used to detect the credibility of a piece of data and explain how our work complements and builds on the state of the art techniques. Credibility is the metric of believability of a statement, action, or source, and the ability of the observer to believe that statement. It is based on the consistency with other evidence [8]. An event observed by different people is usually described in different words. We formalize this observation in the hypothesis section. A rumor can be defined as a talk or opinion widely disseminated with no discernible source cite rumor. Data trustworthiness is the trait of deserving trust and confidence in the data [8]. Trust can be defined as a reliance on the integrity, ability, credibility of a person or source of information [21]. The concept of validity means that information represents real conditions, rules or relationships rather than characteristics of physical objects [4]. Boritz offered a very rich discussion about the data integrity and he includes completeness accuracy and validity as the main characteristics for data integrity [4]. Authoritative is defined as 'clearly accurate or knowledgeable' [1]. Authenticity is defined by 'worthy of acceptance or belief as conforming to or based on fact' [2]. Our contribution can be summarized as designing a creditability model that can rank the credibility of a piece of information. Further, our model is scalable since it is based on PageRank. The importance of the contribution is clear under disaster scenarios where the need of common operating pictures is critical. The existence of several contributers or social media posters results in the need of having a tool that could assess the credibility of the posts. Section 3 presents relevant prior work and section 4 presents our hypothesis and research questions. Section 5 presents the credibility models. We present our system *Credo* in section 6. Section 8 describes our experimental hypothesis and approach. We analyze and discuss the results in section 9. Section 10 closes with the conclusion and further research.

3. RELEVANT WORK

There are several attempts that try to detect for spam and rumor using machine learning algorithms. However, there were very few research endeavors to assess data credibility. We focus here on presenting models of credibility for social networks and especially Twitter. The content of the message can include links, the users mentioned, sentiment of the message and the hash tags used. Posting habits include average number of posts, average posting time between the posts. User characteristics include user centrality, profile creation date. A mix of either the three of the techniques aforementioned or two is sometimes used to quantify or define credibility of a piece of data.

The explosion of data that is disseminated through micro blogging urges for the need to quickly judge the credibility and the trustworthiness of data. Micro blogging platforms

have become a very essential tool to disseminate information. Several research endeavors are trying to detect the credibility of a piece of data. Credibility of information is an important aspect of using information. Data with low credibility is considered of low value. In addition, it is misleading for making decisions. The methods to detect rumors and credible data in literature can fall mainly into three categories: (i)User Characteristics, (ii) posting habits,(iii) message content and (iv) Hybrid. An effective framework should address the set of the aforementioned techniques.

3.1 User Characteristics

The user identity was used to quantify the credibility of data [17]. The assumption presented by Rowe [17] was based on having three layers of identity: real, shared and abstract. The less the gap between the identity layers the higher the probability that the user is credible. As a result, a user who has the three levels coinciding should likely post very highly credible data. The research concluded that a piece of data originating from a user with well known identity on the web is likely to be credible. Although the identity of a user might be useful in tracing his posting habits and judging the credibility of a post, it might not be enough to fully present it. Jin developed a topic initiator detection technique, IRank. The technique can be used to trace information or rumor initiators [10]. IRank was based mainly on the date of the piece of information, originality and centrality of the user. Originator detection can be a step to detect the rate at which the information is disseminated in the network and compare that with the average rate of information dissemination.

3.2 Posting Habits

Several models discussed the techniques of spreading rumors such as Susceptible-Infected-Removed, SIR [22]. In SIR, as people naturally get immune they no longer believe or spread rumors. Small world network enjoys smaller transmission threshold and faster dissemination [13]. Another model presented was the susceptible infected, where an infected user keeps on infecting or sending rumors even when it becomes clear that the piece of information belongs to a rumor campaign. Zhi concluded that warning and controlling can be considered the most important means to stop a rumor [22]. Rumor control and detection were analyzed in [19]. A model for rumor spread in a network was presented where the posting habits of the nodes were characterized. Higher posting rates than the normal were considered a method to spread rumors. The research concluded that as the time required to detect a rumor increases the lifetime of the rumor increases. The model depended on the presence of some non malicious users to detect the normal posting rates and signal the presence of rumors. The methodology presented did not analyze the message content which can be crucial to judge whether a post is meant to spread a rumor.

3.3 Message Content

Schwarz identified several features for credibility and trustworthiness of web pages and data on the Internet [18]. The calculated credibility rank is presented to the user in a visual way to asses the credibility of a page. The model presented was dependent on the domain of the website for example a .gov is more credible than .com and specific news sites are more credible than others. The approach of ranking domains for credibility is useful in judging the credibility of a piece of information. It can be used also as ranking the evidence supplied with a post on a micro blog. However, it should be noted that judging the credibility of a piece of data solely based on the source might not be enough.

3.4 Hybrid

Truthy in [16] followed an approach of detecting the amount of Tweets that are similar originating from an account. The method presented aimed at detecting astroturfing through recording the communication among users and the words they use. In addition, Truthy detected the similarity of messages originating from the same user. The work concluded that high degrees of message similarity is highly correlated with spreading rumors or biasing the opinion of the network. In his work Beck presented a model where he analyzed the message content of Twitter spam and the characteristics of the source users that messages are originating from [3]. Beck showed that the centrality of the user is irrelevant for characterizing the spam. The message content was used to trigger further examination of the message content. Beck concluded that the number of words are too many to result in high quality detection. Carlos aimed at quantifying the credibility of data on Twitter by classifying the Tweets based on user profile, topic and message characteristics [6]. The first phase included manual classification of the Tweets using Amazon Turk as a way to provide a gold metric for the classifier. The classifier used was tree based, J48. The message received a rank of credibility based on several features that fall in the three mentioned categories.

4. HYPOTHESIS AND RESEARCH QUESTIONS

Through our research we try to answer the following research questions in this paper:

1. Given a set of posts how can a variation of PageRank [5] be used to improve credibility detection over a supervised baseline.
2. Given our initial PageRank model, can we improve credibility ranking performance by including information about author posting habits?
3. Based on the observation that some low credibility messages look alike, can we improve performance further by checking semantic versus lexical similarity?
4. Can we improve performance of our models by making them semi-supervised?

We test the hypothesis that an event observed by different people is usually described in different words. Further, different observers describe different aspects of the event with different lexical features. In other words, there is high similarity in the semantic content yet a low level of similarity in lexical features. On the other hand, low credible data has high degree of lexical similarity, low diversity in the number of sources and absence of evidence. We test this hypothesis and extend the findings to verify our model in judging pieces of data. We consider the Tweet as the building unit

of information. A summary of the hypothesis that we address is, an event observed by different observers is described using different linguistic lexicons. As a result, we can find high degree of semantic similarity yet low degree of lexical similarity.

5. CREDIBILITY MODELS

Our Credibility model is based on PageRank. We use the links between messages and posters to define the credibility of a poster and a message. A poster is a user who composes a message and posts on the social media framework. He might have friends or connections. A reporter is a user who have access to view the messages and posts posted by users of a social network. He has the ability to give a score for the credibility of the posts. The user can play more than one role at the same time. However, a user might not play the role of a reporter for the messages composed by him/her. The role of a reporter can be understood as an arbiter who rates the credibility of a message. In case the system does not support the concept of a reporter, manual ranking of message credibility will be required to apply our model. In this section, we introduce several instantiations of our framework where we experiment with different features and parameters to reach the optimal performance. Subsection **??** introduces our Reporter-Message model using messages and reporters. In subsection 5.3 we develop a model of Poster-Reporter-Message. Section 6 presents a model of message reporter and top k similar messages. In this model, the credibility of a message is calculated based on reporter trustworthiness, poster credibility and the credibility of the top k semantically similar messages. We use the trustworthiness for persons while credibility is for messages, data or information.

5.1 Credo PageRank Algorithm

In [15] Page, Brin et. al. described an approach for the estimation of the importance of a web page based purely on the link structure of the world wide web. Their proposed score PageRank was based on the assumption, that a document spreads his relevance equally to all documents to which it links. We extend this technique and use it to calculate the credibility of a message. There are three types of nodes reporters, posters and messages. We do not make a distinction while running the algorithm about that. Each of these nodes is expected to carry a credibility value that is converging to the final value of credibility. The goal is to calculate the credibility of the messages. We use the Algorithm presented with the appropriate equation according to the model under computation to calculate the message credibility. In the model of reporter-message only reporters and messages are included in the computation. However reporter-poster-message include three types of nodes.

5.2 Reporter Model

In the *Reporter Model* we have two types of nodes in the graph the reporters rank the credibility of a message in three categories rumor, low credibility and credible. The categories mentioned before are then converted into a number scale. Our algorithm calculates two scores one for message credibility and the other for poster trustworthiness. The credibility score for a given message is the sum of all the reporters score.

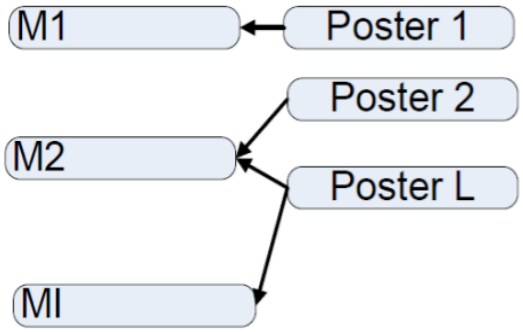


Figure 1: Reporter Model

$$MessageCredibility = \sum_{i=1}^{n} S(R_i) \tag{1}$$

$$PosterTrustWorthiness = \sum_{i=1}^{m} M(S_i) \tag{2}$$

$$ReporterTrustWorthiness = \frac{num}{\sum_{i=1}^{num} |M(S_i) - M(S_g)|} \quad where |M(S_i) - M(S_g)| > 0 \tag{3}$$

The reporter trustworthiness is defined by the inverse distance between the real value for credibility of messages rated by the reporter and the gold metric. A simplification for the model is to consider all reporters with the same trustworthiness value. In our experiments we considered posters with the same credibility value. Equation 3 describes the inverse summation of the difference in message credibility between the gold metric credibility value and the message credibility value reported by a reporter. In equation 3 the value of zero for the denominator will trigger an infinity value for the credibility; as a result, we define the function $ReporterTrustWorthiness$ as a continuous function defined by the equation 3 where the value of the function equals $Cred_{Max}$ where $Cred_{Max}$ is initialized to a value of 5. Which is the maximum credibility value of messages in our framework. The summation is normalized by num, the total number of messages for the summation the poster rated. The reporter trustworthiness can be defined as the summation of the real credibility value of the messages rated by such reporter. Examples of social networks that adopt a reporter include *Youtube* and *Facebook*. However, *Twitter* does not have this feature where the users can rank the post or give a feedback about it.

The poster credibility is calculated based on the credibility of the messages which can be manually assigned or rated by human evaluators; this is the supervised part where we rely on having at least one of the three entities (messages, reporters or posters) to have an initial valid credibility score that reflects the real values of the credibility measures to

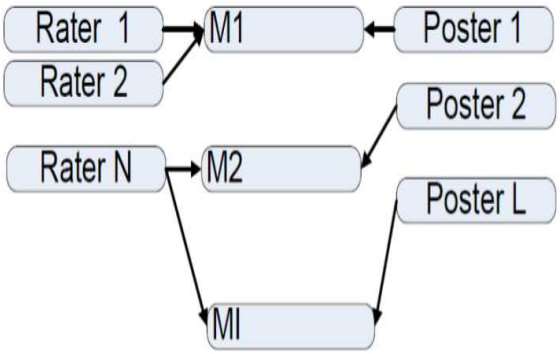


Figure 2: Credo Semi-Supervised Poster Reporter Model

start computing the values of the other two entities recursively. Equations 1 and **??** define the credibility of the poster and of the message recursively. The credibility of a message and the trustworthiness of a reporter are calculated through an iterative procedure. We initialize the reporter scores uniformly with values from 0 to 1. A credibility score for a message is calculated using the reported scores in the previous iteration. This process is repeated until the required convergence is reached. We set convergence value to 0.00001.

5.3 Poster Reporter Message

$$MessageCredibility_i = \alpha * PosterTrustworthiness_{i-1} + \beta * ReporterTrustworthiness_{i-1} + \gamma * MessageCredibility_{i-1} \quad (4)$$

We extend the first model by adding the poster of the message and the credibility of a message composer. Initially the credibility of a poster is defined by the number of credible messages he posts. Note that α, β and γ are scaling factors; we set the following values to adjust the effect of each component in message credibility, $\alpha = 0.35$, $\beta = 0.4$ and $\gamma = 0.25$ with summation of 1.0. The scaling factors can receive values between 0 and 1.0, where $\alpha + \beta + \gamma = 1.0$. Equation 4 describes how the message credibility is computed. Every iteration the poster trustworthiness reporter trustworthiness and previous message credibility value are computed accordingly. The first time the values are computed using the initial scores until the maximum required iteration or until we reach a convergence value . By changing the values of the scaling factors we can give more weight to either the historical or the new coming information.

6. CREDO: SEMI-SUPERVISED MODEL

This model develops links between similar messages to assess the credibility value. We consider feature and semantic similarity. Subsection 6.1 explains the feature selection in more detail. We chose the top K most similar messages to the message at hand to extract evidence for an event. We experimented with $n = 5$ and $n = 10$. We found that the model gave better results when using the top 5 similar messages. We present the results of the top 5 similar messages. In this model each iteration performs a weighted summation for the reporter trustworthiness, the poster credibility and the credibility of the top k semantically similar messages. We set a threshold for considering the messages, in case there are less than 5 semantically similar messages that are relevant to the current post inspected, only the messages with a semantic score more than the threshold are considered in this summation.In the first iteration the initial scores are used for the three entities then the merssage creiblity is calculated recursively and iteratively until the maximum numeber of iterations is reached or the required conversion or error factor is reached.

$$MessageCredibility_i = \alpha * PosterCredibility_{i-1} + \beta * Reportertrustworthiness_{i-1} + \gamma * MessageCredibility_{i-1} + \zeta * \frac{\sum_1^k M(S_i)}{k} \quad (5)$$

Note that the absence of one of the terms above sets the value to zero which can affect the accuracy of the model yet it still can provide a credibility metric. The reason behind adding similar messages is that messages with similar content should have similar credibility score as a result credibility scores should be propagated between messages. However, content similarity only might be misleading. Adding the other features presented in subsection 6.1 improves the credibility scoring framework; more on this is in the discussion section. Equation 6 was used to calculate the average convergence of the credibility schemes developed.The formula subtracts the current credibility value for the message from the calculated value in the previous iteration. The formula is run for the whole number of messages in the system then the value accumulated is divided by the total number of messages.

$$AverageConvergence = \frac{\sum_{i=1}^{i=iterations}((|C_{i-1} - C_i|)/C_i)}{num} \quad (6)$$

C_i: Credibility value for a message at the i^{th} iteration. C_{n-1}:Credibility of the message at the $(i-1)^{th}$ iteration. num: total number of messages.

We introduced unsupervised credibility evaluation models. We extend the models to be semi supervised. The semi supervised scores are the scores given by the evaluator. We designed the semi-supervised variants of our models by fixing the credibility score for a message. We initialize the credibility values for our models uniformly before we start calculating recursively with values from 0 to 1. The credibility score for the surrounding nodes were calculated using Equation 5. A message node M has a credibility value C which is assigned a value 5. In other words the score $(M) = 5$ for credible messages. On the other hand, a message that has low credibility score $(M) = 0$ and a message that is considered a rumor has a score of score $(M) = -5$.

6.1 Feature Selection

We build on the work of [7] by using some of the features used in his work and we add some more features that aid

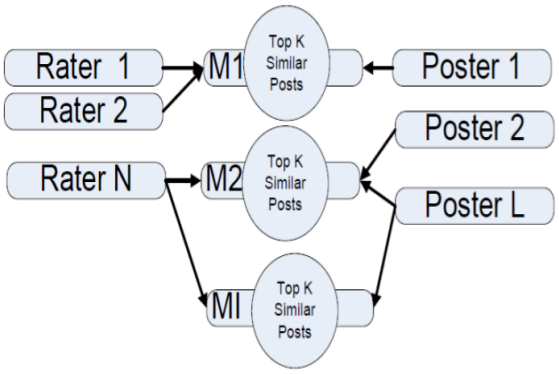


Figure 3: Credo: Semi-Supervised Poster Reporter Top K Messages

our model to improve its precision. The features used are used to cluster messages with similar credibility features. Ratio of lexical similarity to semantic similarity: we developed a semantic module that takes two posts and map them to Wikipedia articles and gives a score of semantic similarity according to the angle between the two posts. We also measure the lexical similarity between two posts by calculating the angle between them. Wikipedia has grown to be the world largest and busiest free encyclopedia, in which articles are collaboratively written and maintained by volunteers online. Wikipedia has been successful as a means of knowledge sharing and collaboration [9]. Average number of postings for user: We get the average number of posts per week for a poster. Evidence: we check whether the post has a url that can help as evidence for the post and the data contained. Ratio of verbs and adjectives to nouns: We calculate the ratio between verbs and adjectives to nouns. Absence or presence of exclamation and question mark. Sentiment level: We calculate the sentiment level for the post. We use the framework we developed in [14] Hash tags: this helps to group posts that discuss the same topic.

7. SEMANTIC MAPPING

A string T consists of n words $T = l_1...l_i$. A *Wikipedia* dump $W = w_1, ...w_n$ of n articles are indexed as an index I. Every Tweet T is represented as a vector $V = t_1, ...t_i$ where i is the total number of the terms in a Tweet. We measure the similarity between the two Tweets by using the cosine and Jacquard to compute the string similarity. After mapping the posts to articles in wikipadiea we calculate the similarity vector between the top most similar articles to the Tweet vectors. We follow an approach similar to the one presented in [11] to calculate the most semantically similar messages. We chose two standard similarity measures, the cosine and the jaccard, illustrated by equation 7 and 8 respectively. The results presented are based on the cosine.

$$cos(\Theta) = \frac{|T_1|.|T_2|}{||T_1||.||T_2||} \tag{7}$$

$$Jaccard(T_1, T_2) = \frac{|T_1 \bigcap T_2|}{|T_1 \bigcup T_2|} \tag{8}$$

8. EXPERIMENTS

Our data set consists of twitter messages and credibility reports. The Tweets were collected using Twitter API the Rest. Each credibility report is associated with one message. The data was collected on 5th January. The total number of messages was 1087 collected. The messages collected are in English. We filtered the messages that have no hash tags. In addition, we filtered the messages that have hash tags that occurred less than 10 times in the entire set of the data collected. We removed the stop words used by using the stop words of Lucene [1]. We used three independent raters to annotate the message credibility of the message. We asked for the annotation of 3 independent raters who are familiar with using twitter. A message is considered an advertisement if it is reported by one of the raters as a promotional advertisement or commercial. For example, it can be a commercial message or a request for somebody to follow another user. These types of messages are considered irrelevant to our model and as a result, we excluded them. In addition, every message can be classified in one of the three categories (rumor, low credible, credible). The promotional messages are discarded and the average score of the three raters is computed and considered the credibility score of the post. The raters annotated 233 messages to be of low credibility, 124 to be commercial, 116 as rumors and 613 messages as credible. We use 25% of the set for training and 75% for testing. As a result the older messages were used to train the models. We also applied five folds and we selected 25% of the posts to randomly train the models and we tested with the 75%. The results presented are the average of the 5 folds. We used the same test set for both the supervised and semi-supervised models. We kept the ratio of 25% for all the messages that is, the rumors, the low credibility messages and the credible ones. The margin of error for the results presented are within the 95% confidence interval. Both the chronological set of messages and the folds led to similar results. We are discussing and presenting the folds part of the experiment.

8.1 Baseline

We compare our credibility models to two baselines, the first is independent of rater reports and it uses text as an evidence of credibility level. On the other hand, the second baseline uses the ratings of the raters.

8.1.1 Naive Bayes Classifier

This baseline uses the Naive Bayes classifier on the text content of messages. We use the bag of lexicons of a Tweet after removing the stop words as a feature. We extend the use of Bayesian networks for rumor detection and credibility ranking.

9. RESULTS AND DISCUSSION

This section presents the results of our models. We start by comparing these results to the our reporter poster k messages model. We compare the models presented according to their ability to rank and spot messages according to the credibility level. We record the Precision and Recall and calculate the F-measure for all the models presented. Table 1 shows the area precision recall and the F-measure for the models tested.

[1] `http://lucene.apache.org/`

	Bayesian Networks	Reporter-Message	Reporter-Poster-Message	Reporter-Poster-Top-K
Recall	0.82	0.8	0.87	0.95
Precision	0.7	0.85	0.88	0.93
F-measure	0.75	0.844	0.874	0.93

Table 1: Precision recall, F-measure

Eighty seven percent of the low credible messages had a ratio of less than 0.35 for the Lexical to semantic similarity. On the other hand, 81.5% of the credible posts had a value greater than 0.65 for the same ratio. On the other hand 68% of the messages classified as rumors had no evidence or url associated with them. In addition, we can associate that the high lexical similarity was strongly correlated to the presence of low credibility of data, which can trigger the need for further investigations of the messages and their content. The Reporter top k message Model gave the best performance of our three rumor detection and credibility detection models.

9.1 Processing Updates

Dealing with updates is one of the essential design caveats in our framework. There are usually two approaches of dealing with updates: calculation from scratch and online updating. In the first approach the new result is calculated from scratch where the algorithm needs to have access to all the historical data required to calculate the new value of credibility for all the players in the system. However, this technique is expensive though it can offer exact solutions or accurate ones. This situation leads to an accurate framework that might not scale well especially with huge amounts of data. On the other hand, the online method of updating the credibility of a message can be faster and more efficient. However, the accuracy might be affected. We designed a pilot experiment to measure the quality of the online updating technique. We start with 50 % of the total number of messages then we grew until the system reached 100 % of the total number of initial messages and posters. The increments were of 10%. The value of the credibility for the old messages is initiated with the last value computed. The newly arrived messages their values are randomly initialized if they had no previous values by the reporters. The framework was run to reach the the convergence required. On average we needed to run 27 iterations until we reach the convergence requested. This is compared to running on average 170 iterations to reach the same level of convergence. As a result, online updating is more efficient. However, further investigation for this set of experiment should reveal other caveats in the technique we developed.

10. CONCLUSION AND FUTURE WORK

We presented three models for credibility estimation. The concept of credibility has become an important characteristic for useful data. Quantifying and formalizing credibility is an essential stage in developing systems that can measure credibility of pieces of information. The three models presented perform better than the baseline over the set of messages and values measured. The models we presented are scalable models in nature since there are several parallel implementations for the PageRank. Measuring semantic similarity proved to be a useful feature to measure credibility of a post in addition to detecting the absence or the presence of evidence. An important aspect that we plan to carry out more research on is the collusion of reporters to skew the results of the system. We plan to test the minimum percent of reporters that are needed in order to receive a specific degree of confidence and accuracy. This can lead us to extending the framework and incorporating game theory rules that can include incentives for good reporters and punish misbehaving ones.

11. ACKNOWLEDGEMENTS

This work has been supported in part by the Carnegie Mellon University Silicon Valley Disaster Management Initiative and by a DMI affiliate, IntraPoint. The work has also been supported partially by Amazon grant. The work has also been supported by IMT Lucca, Italy.

References

[1] Merriam webster. `http://www.merriam-webster.com/dictionary/authoritative`, 2011.

[2] Merriam webster. `"http://www.merriam-webster.com/dictionary/authoritative"`, 2011.

[3] K. Beck. Analyzing tweets to identify malicious messages. In *Electro Information Technology (EIT), 2011 IEEE International Conference on*, pages 1–5. IEEE.

[4] J. Boritz. *Managing enterprise information integrity: security, control, and audit issues*. Isaca, 2004.

[5] M. Brinkmeier. Pagerank revisited. *ACM Trans. Internet Technol.*, 6(3):282–301, Aug. 2006.

[6] C. Castillo, M. Mendoza, and B. Poblete. Information credibility on twitter. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 675–684, New York, NY, USA, 2011. ACM.

[7] C. Castillo, M. Mendoza, and B. Poblete. Information credibility on twitter. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 675–684, New York, NY, USA, 2011. ACM.

[8] J. G. Conrad, J. L. Leidner, and F. Schilder. Professional credibility: authority on the web. In *Proceeding of the 2nd ACM workshop on Information credibility on the web*, WICOW '08, pages 85–88, New York, NY, USA, 2008. ACM.

[9] M. Hu, E. Lim, A. Sun, H. Lauw, and B. Vuong. Measuring article quality in wikipedia: models and evaluation. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 243–252. ACM, 2007.

[10] X. Jin, S. Spangler, R. Ma, and J. Han. Topic initiator detection on the world wide web. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 481–490, New York, NY, USA, 2010. ACM.

[11] C. Lucchese, S. Orlando, R. Perego, F. Silvestri, and G. Tolomei. Detecting task-based query sessions using collaborative knowledge. In *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pages 128–131. IEEE, 2010.

[12] M. McCord and M. Chuah. Spam detection on twitter using traditional classifiers. In J. Calero, L. Yang, F. MÃąrmol, L. GarcÃ■a Villalba, A. Li, and Y. Wang, editors, *Autonomic and Trusted Computing*, volume 6906 of *Lecture Notes in Computer Science*, pages 175–186. Springer Berlin / Heidelberg, 2011.

[13] C. Moore and M. Newman. Epidemics and percolation in small-world networks. *Physical Review E*, 61(5):5678, 2000.

[14] A. Nagy and J. Stamberger. Crowd sentiment detection during disasters and crises. In *Proceedings of the 9th International ISCRAM- International Conference on Information Systems for Crisis Response and Management*, volume 1, 2012.

[15] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. 1999.

[16] J. Ratkiewicz, M. Conover, M. Meiss, B. Gonçalves, S. Patil, A. Flammini, and F. Menczer. Truthy: mapping the spread of astroturf in microblog streams. In *Proceedings of the 20th international conference companion on World wide web*, WWW '11, pages 249–252, New York, NY, USA, 2011. ACM.

[17] M. Rowe. The credibility of digital identity information on the social web: a user study. In *Proceedings of the 4th workshop on Information credibility*, WICOW '10, pages 35–42, New York, NY, USA, 2010. ACM.

[18] J. Schwarz and M. Morris. Augmenting web pages and search results to support credibility assessment. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, pages 1245–1254, New York, NY, USA, 2011. ACM.

[19] R. M. Tripathy, A. Bagchi, and S. Mehta. A study of rumor control strategies on social networks. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 1817–1820, New York, NY, USA, 2010. ACM.

[20] A. H. Wang. Don't follow me: Spam detection in twitter. In *Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on*, pages 1 –10, july 2010.

[21] S. Young and J. Palmer. Pedigree and confidence: Issues in data credibility and reliability. In *Information Fusion, 2007 10th International Conference on*, pages 1 –8, july 2007.

[22] Z. Zhu and D. Liao. Research of rumors spreading based on transmission dynamics of complex network. In *Management and Service Science (MASS), 2010 International Conference on*, pages 1 –4, aug. 2010.

Automatic multi-label categorization of news feeds

Majid Darabi*, Hossein Adeli*, Nasseh Tabrizi**
Department of Computer Science
East Carolina University
East Fifth Street, Greenville, NC 27858-4353 USA
*{Darabim10, Adelijelodarh10}@students.ecu.edu 252-328-9626
**Tabrizim@ecu.edu 252-328-9691

Abstract— **Web feeds play an important role for publishing information on Internet. On one side, News feeds, special type of web feeds, are used by the news websites to publish their news effectively. On the other side, feed readers help the users to keep up with large amount of news feeds to which they are subscribed. Using Library of Congress Subject Heading to expand the user's predefined labels, we present fully automatic RSS formatted news feed categorization system in this study. We propose a ranking model to find the relevancy of feed items to the category labels. To evaluate our method, we developed a feed reader and conducted quantitative experiments that show effectiveness of the method.**

Keywords: RSS feed, news categorization, learning to rank, query expansion, LCSH

I. INTRODUCTION

In recent years, the number of news, articles, and information published on the web has grown dramatically. This excessive growth has made it too difficult for individuals to keep up with the pace of published information and frequency of their updates. To address this issue, websites publish information using web feed or syndicated feed to provide users with frequently updated content.

Really Simple Syndication (RSS), a type of web feed, is an XML document that facilitates content syndication [1]. News websites such as BBC, CNN and Reuters publish daily news using RSS format in a form of news feed. Every news feed consists of a root (<rss>) and one <channel> element. A <channel> contains metadata about news and at least one news' <item>. <item> contains three core elements that characterizes postings or articles. <title> is the headline of the article that is usually picked very general and short. <description> element mostly contains summary or opening sentences of the news body. Each news item is linked to the original article via a <link> element.

Publishing news in form of web feeds allows users to subscribe to their favorite news provider through *feed readers* or *aggregators*. A feed reader is supplied with links to the news provider and it can check the subscribed feeds to retrieve newly added or updated news. Feed readers are the solution to the problem of gathering all the relevant news but for a user finding interesting news among explosive number of feeds is a tedious task. Categorizing news feed can help the user to find favorite news easily and avoid spending much time to search among many news items. News categorization is the process of assigning appropriate user generated predefined label to a news article.

In this manuscript, we propose a categorization method called LabeledNews to categorize streaming news items. The LabeledNews categorizes news items directly from news feeds without retrieving original content. Our method is based on query expansion using a *thesaurus* to categorize news feeds conceptually to predefined user's labels. Additionally, the LabeledNews sorts news items based on their relevancy in each category.

The rest of the paper is organized as follows. Section II introduces related work. The details of LabeledNews algorithm which includes query expansion approach and ranking function are shown in Section III. In section IV, we present evaluation of our method based on experimental results. Finally, we conclude the paper in section V.

II. RELATED WORK

Many studies have been conducted to improve news feed categorization by using different measures of text similarity. Wegrzyn-Wolska *et al.* [6] proposed a classification method based on fuzzy similarity measure. PeRSSonal [18] is a feed reader which categorizes RSS feeds based on the cosine similarity measure, dot products and term weighting calculations. Personalized News Categorization [16] was proposed in order to classify the articles in a 'per-user' manner. The system uses a classification technique that represents documents using the vector space representation of their sentences.

The news item usually does not include entire article but only title and small description. Therefore they do not have enough word co-occurrence or context shared information for effective similarity measures [16, 17]. So most of the conventional categorization algorithms may fail when directly applied to news item [5]. For the same reason, just searching label's terms inside the news item would not be a very effective approach for finding the relevancy of news items.

Fig. 1, shows two sample news items from BBC news feed. First you can see those news items contain partial information about original articles. Second even though the

```
<item>
    <title>Gardeners told 'wash off compost'</title>
    <description>Gardeners are warned to wash their hands after using compost following a series of Legionella cases in Scotland over the past five years.</description>
    <link>http://www.bbc.co.uk/news/uk-scotland-18206191#sa-ns_mchannel=rss&amp;ns_source=PublicRSS20-sa</link>
    <guid isPermaLink="false">http://www.bbc.co.uk/news/uk-scotland-18206191</guid>
    <pubDate>Mon, 28 May 2012 00:40:37 GMT</pubDate>
</item>
```

News-item-1

```
<item>
    <title>Birth 'neglect' factor in death</title>
    <description>A gross failure to provide basic medical attention to a mother in childbirth at Wales' biggest hospital contributed to her baby's death, a coroner finds.</description>
    <link>http://www.bbc.co.uk/news/uk-wales-south-east-wales-18235127#sa-ns_mchannel=rss&amp;ns_source=PublicRSS20-sa</link>
    <guid isPermaLink="false">http://www.bbc.co.uk/news/uk-wales-south-east-wales-18235127</guid>
    <pubDate>Mon, 28 May 2012 18:30:26 GMT</pubDate>
</item>
```

News-item-2

Fig. 1. Two sample news items from BBC

news items 1 and 2 belong to the "health" category, using just the term "health", we cannot categorize news-item-1 as a member of "health" category. Also notice applying traditional similarity measures such as cosine coefficient [13, 14] would not be effective.

III. LABELEDNEWS

A. The Label Expansion

Category labels defined by users are usually one or two terms; more like subject headings that captures the essence of interesting topics to the user. Given that and the fact that news item only includes short description and title, searching for relevant news feed to the user defined labels suffers from the typical mismatch issue. Query Expansion methods have been examined to effectively solve this issue by enriching the original query using external source of information [5,7,8]. These methods expand the query by adding relevant terms that are likely to appear in relevant documents [5]. One of the effective sources of information is human or computer generated thesaurus [9].

In this study, we employ the Library of Congress Subject Headings (LCSH)[1] to expand the labels. The Library of Congress Subject Headings (LCHS) is by far the most widely adopted subject indexing language [4]. Subject headings are organized conceptually as controlled vocabulary so that every subject is described by a single term. Each subject heading has 3 types of relevant terms; Related Terms (RT), Broader Terms (BT) and Narrower Terms (NT). Fig. 2, shows the subject heading "Sport" and few of its relevant terms. "Recreation" is a BT for "Sport", "Ball Games" and "Aquatic Sport" are narrower terms of "Sport" and "Game" and "Athletic" are two related terms for "Sport".

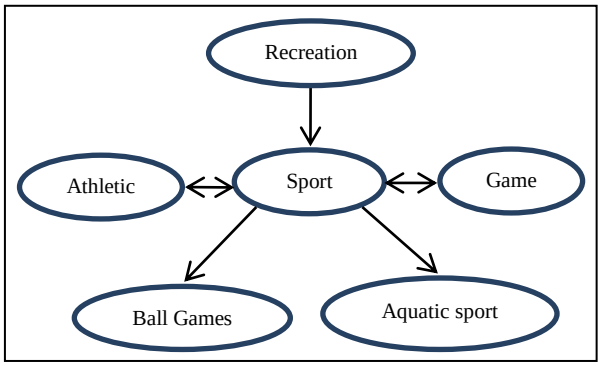


Fig. 2. "Sport" Related terms in LCSH

B. Categorization Approach

For any arriving news item, LabeledNews computes the relevancy of all predefined categories to that news item. As mentioned the news item only includes a short part of the original article, consequently there are a limited number of word co-occurrences and not enough discriminative clues; therefore it does not fall into one conceptual category with ac-ceptable accuracy. Considering that, the LabeledNews is designed to be a multi-label categorization [15] system that takes the ranking categorization approach as opposed to hard categorization policy [11]. In hard categorization, the classifier makes a decision to assign only one category to every input instance.

Ranking categorization sorts categories based on their estimated relatedness to a given document. In this method, the greater ranking value of a category for a document indicates higher relevance between them. The LabeledNews employs a Thresholding strategy [16] to determine the number of categories that will be assigned to a news item. Three common strategies are RCut, PCut and SCut [17, 5] from which we have used RCut for evaluation purpose.

C. Ranking Function

We have developed a ranking function $\boldsymbol{R}: I \times L \rightarrow \mathbb{R}$ to compute the relevancy score between news item $i_k \in I$ and category label $l_j \in L$ which is denoted by $\hat{r}_{kj} = \boldsymbol{R}(i_k, l_j)$.

Before starting feature selection, we select a subset of available elements in a news item to get rid of useless information and speed up computations without loss of performance. Shin and Park [2] have explored how selecting different subsets of news feeds' elements affect the classification performance. In this study, we consider contents of <title>, <description> and <link> elements for each news item; because these elements contain information which describes the news and also help to individuate news

[1] http://id.loc.gov/authorities/subjects.html

items. Link elements are usually informative as they include words related to the article.

We use the bag of terms model [3] to represent news item's title, description, and link elements in vector space model [13]. For every news item's description (d) and every label (l), we employ normalized term frequency method [12] to compute the descriptions feature score for that label ($\delta(d,l)$) as follows:

$$\delta(d,l)=\frac{1}{|d|}\sum_{t\in l} tf(d,t) \qquad (1)$$

In (1) $|d|$ represents the total number of terms in d and $tf(d,t)$ is the number of occurrence of term $t\in l$ in document d . In order to compute feature scores of news item's title and link elements (lt) and a label, we use term frequency method defined as:

$$\sigma(lt,l)=\sum_{t\in l} tf(lt,t) \qquad (2)$$

Relevancy score between news item i_k and label l_j , $\boldsymbol{R}(i_k,l_j)$, is the weighted sum of individual feature scores that are computed using (1) and (2). We expand every label l_j using their RTs, BTs and NTs from LCSH. So there are four different set of terms for every label and since we have considered three elements of each news item (title, description and link elements); we end up with the total number of twelve of feature scores. In other words, in vector space model, each news item is represented as a twelve dimensional vector corresponding to a label. The ranking function $\boldsymbol{R}$ has twelve weights associated with these twelve feature score functions.

In order to train the model, we need a training data which is a set of tuples (news item, relevancy vector), and the relevancy vector shows the news item's degree of relevancy to all labels. Table I, shows the training process.

For an arriving news item i_k , we will calculate the feature vector, $\phi(i_k,l_j)$, for each label l_j . Then the relevancy of i_k to l_j is computed as follows (using trained weights vector w): $\boldsymbol{R}(i_k,l_j)=w^T.\phi(i_k,l_j)$. After calculating the relevancy scores, LabeledNews employs a strategy to assign few top relevant labels to i_k .

IV. Experiments

To evaluate the LabeledNews methodology, we present quantitative results using our web based feed aggregator developed by the authors. Table II, shows the source of news feeds to which the aggregator was subscribed and the crawler gathered 1366 news items in 6 days. We selected 17 labels for training and another 12 labels for testing. Five human experts were assigned to manually grade the relevancy between all labels and news items. The grades were from the following set: {extremely relevant, relevant, low relevant, not relevant}.

Table I
Training process of LabeledNews method

Input:

Training set of tuples <news item i_k , relevancy vector RV_k >

News item i_k : title i_k^t , description i_k^d and link i_k^l

Process:

1. For every predefined label $l_i \in \{l_1,l_2,\ldots,l_n\}$
   1.1. Using LCSH find l_i 's RT (l_i^{RT}), NT (l_i^{NT}) and BT (l_i^{BT}) terms
2. For the news item $i_k \in$ training set
   2.1. For every predefined label $l_i \in \{l_1,l_2,\ldots,l_n\}$
      2.1.1. Using (2) calculate the i_k^t feature scores
      ($\sigma(i_k^t,l_i), \sigma(i_k^t,l_i^{RT}), \sigma(i_k^t,l_i^{NT}), \sigma(i_k^t,l_i^{BT})$)
      2.1.2. Using (1) calculate the i_k^d feature scores
      ($\delta(i_k^d,l_i), \delta(i_k^d,l_i^{RT}), \delta(i_k^d,l_i^{NT}), \delta(i_k^d,l_i^{BT})$)
      2.1.3. Using (2) calculate the i_k^l feature scores
      ($\sigma(i_k^l,l_i), \sigma(i_k^l,l_i^{RT}), \sigma(i_k^l,l_i^{NT}), \sigma(i_k^l,l_i^{BT})$)
      2.1.4. Define feature vector

$$\phi(i_k,l_j)=\begin{pmatrix}\sigma(i_k^t,l_i), \sigma(i_k^t,l_i^{RT}), \sigma(i_k^t,l_i^{NT}),\\ \sigma(i_k^t,l_i^{BT}), \delta(i_k^d,l_i), \delta(i_k^d,l_i^{RT}),\\ \delta(i_k^d,l_i^{NT}), \delta(i_k^d,l_i^{BT}), \sigma(i_k^l,l_i),\\ \sigma(i_k^l,l_i^{RT}), \sigma(i_k^l,l_i^{NT}), \sigma(i_k^l,l_i^{BT})\end{pmatrix}$$

      2.1.5. Define <feature vector, relevancy score> as
      $\langle \phi(i_k,l_j), RV_{k,j} \rangle$
1. Train weights ($\vec{w}=[w_1,w_2,\ldots,w_{12}]$) of mapping $\boldsymbol{R}=w^T.\phi$ from feature vectors to relevancy vector.

Output:

Trained weights w

Table II
News Feed sources

http://www.bbc.co.uk/news/business
http://www.bbc.co.uk/news/health
http://www.bbc.co.uk/news/science_and_environment
http://www.bbc.com/news/technology
http://www.bbc.co.uk/sport/0
http://money.cnn.com/news/economy
http://www.cnn.com/POLITICS
http://www.cnn.com/TECH
http://www.cnn.com/HEALTH
http://www.reuters.com/politics
http://www.reuters.com/news/technology
http://www.reuters.com/news/sports
http://www.reuters.com/news/health

To train the model, we have used RankSVM method [10] which is a widely accepted and highly effective ranking method. The thresholding strategy that we have used, to each news item, assigns only the labels which score higher relevancy than certain percentage of the most relevant label which we call Thresholding Percentage (TP).

We present the 11-point interpolated average precision measure of LabeledNews method in Figure 3. We evaluated the method for three different Thresholding Percentages and compare them to the case of not expanding labels.

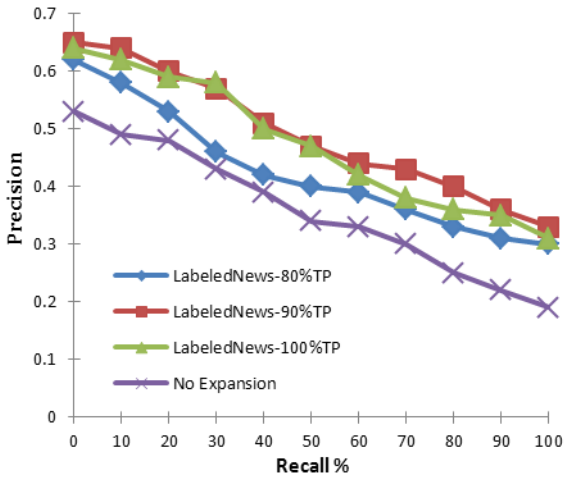


Fig. 3. Performance of label expansion and original label methods

As shown in Fig. 3, Using LCSH to expand labels results in significant improvement in Recall. The LabeledNews also improves the precision. Among different Thresholding Percentages, 90% proved to have highest performance.

Table III
Micro, Macro and F1 measures

Measure (%) / Method	Micro Averaging			Macro Averaging		
	Prec	Rec	F_1	Prec	Rec	F_1
LabeledNews-100%TP	62.7	68.2	66.3	62.6	68.2	66.2
LabeledNews-90%TP	60.1	75.5	66.9	60.1	76.4	67.3
LabeledNews-80%TP	55.9	78.3	65.2	55.4	78.5	64.9
No Expansion	52.8	38.4	44.4	52.8	38.6	45.1

Table III, illustrates micro and macro averaging and F_1 measures for LabeledNews categorization results. Comparing with No Expansion method, we observe that labeledNews has significantly higher recall but precision doesn't improve dramatically. The calculated F_1 measure in both micro and macro averaging has improved.

V. Conclusion

In this paper, we presented a method for news feed categorization. We used related, broader and narrower terms in Library of Congress Subject Heading to expand each predefined label term and developed a ranking function to compute the relevancy score between a news item and a label. The experiments show that our method significantly improves the categorization quality compared to using the original label without expansion.

References

[1] K. E. Gill, Blogging , RSS and the Information Landscape : A Look At Online News. *Wall Street Journal*, (2005).W.-K. Chen, *Linear Networks and Systems* (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.

[2] Y. Shin, and J. Park, "An SVM Based Approach to Feature Selection for Topical Relevance Judgement of Feeds," *Workshop of the 33rd Annual International ACM SIGIR Conference*, 2010, p.40-43.

[3] C. D. Manning, P. Raghavan, and H. Schutze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[4] J. D. Anderson, and M. A. Hofmann. "A fully faceted syntax for *Library of Congress Subject Headings*," *Cataloging & Classification Quarterly*, v. 43, no. 1 (2006), p. 7-38.

[5] D. Metzler, S. Dumais, and C. Meek, "Similarity measures for short segments of text," *In Proceedings of the 29th European conference on information retrieval (ECIR 2007). Lecture notes in computer science*, vol 4425, Springer, pp 16–27, 2007.

[6] k. Wegrzyn-Wolska, and P. S. Szczepaniak, "Classification of RSS-Formatted Documents Using Full Text Similarity Measures," (L. D & G. M, Eds.)*Lecture Notes in Computer Science*, *3579*, 400-405, 2005.

[7] V. Lavrenko, W. B. Croft, "Relevance based language models," *In: Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval*, New Orleans, Louisiana, September 9–13. ACM Press, New York, pp 120–127, 2001.

[8] Zhai C, Lafferty J (2001) "Model-based feedback in the language modeling approach to information retrieval," *Proceedings of the tenth international conference on Information and knowledge management*, Atlanta, Georgia, October 5–10. ACM Press, New York, pp 403–410.

[9] JING, Y. AND CROFT, W. B. 1994." An association thesaurus for information retrieval." In *Proceedings of the Intelligent Multimedia Information Retrieval Systems* (RIAO '94, New York, NY), 146–160.

[10] T. Joachims," Optimizing Search Engines Using Clickthrough Data," *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining* (KDD), ACM, 2002.

[11] F. Sebastiani, "Machine Learning in Automated Text Categorization." *ACM Computing Surveys*, *34*(1), 1-47. ACM, 2001.

[12] Y. H. Li and A. K. Jain, "Classification of text documents," *The Computer Journal*, vol. 41(8), pp. 537-546, 1998.

[13] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, 18:613-620, 1975.

[14] G. Salton and M. J. McGill. Introduction to Modern Information Retrieval. McGraw-Hill Book Company, 1983.

[15] G. Tsoumakas, I. Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining* 3, 1–13, 2007.

[16] X. H. Phan, L. M. Nguyen, and S. Horiguchi, "Learning to classify short and sparse text & web with hidden topics from large-scale data collections," *In Proc. WWW* Beijing, China, , 91-100. 2008.

[17] X. Hu, N. Sun, C. Zhang, and T. Chua, "Exploiting internal and external semantics for the clustering of short texts using world knowledge," *CIKM* 919–928, 2009.

[18] C. Bouras, V. Poulopoulos, and V. Tsogkas, "Adaptation of rss feeds based on the user profile and on the end device," *J. Netw. Comput. Appl.*, 33(4):410--421, 2010.