

# AusDM04

AI  
2004  
Complex  
2004

## Proceedings 3<sup>rd</sup> Australasian Data Mining Conference

6 - 7<sup>th</sup> December, 2004, Cairns, Australia

Edited by  
Simeon J. Simoff and Graham J. Williams

---

in conjunction with  
the 17th Australian Joint Conference on  
Artificial Intelligence AI2004  
and  
the 7th Asia Pacific Conference on  
Complex Systems COMPLEX 2004

---



University of Technology Sydney  
2004

© Copyright 2004. The copyright of these papers belongs to the paper's authors. Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage.

Proceedings of the 3<sup>rd</sup> Australasian Data Mining Conference – AusDM04, 6 – 7<sup>th</sup>, December, 2004, Canberra, Australia, in conjunction with the 17th Australian Joint Conference on Artificial Intelligence AI2004 and the 7th Asia Pacific Conference on Complex Systems COMPLEX 2004.

S. J. Simoff and G. J. Williams (eds).

Conference Web Site:

<http://www.togaware.com/ausdm04/>

Published by the University of Technology Sydney

ISBN 0-646-44379-8

## Foreword

The Australasian Data Mining Conference series, initiated in 2002 and known as **AusDM**, has become the annual flagship event of the data mining and business analytics researchers and industry practitioners in the region. The conference series has a unique profile in nurturing this joint community. The first and second edition of the conference (held in 2002 and 2003 in Canberra, Australia, in conjunction with the 15th Australian Joint Conference on Artificial Intelligence and the Congress on Evolutionary Computation, respectively) attracted participants from Australian industry, academia, and research institutions and centres. These meetings facilitated the links between different research groups in Australia and industry, evidenced by the initiation and creation of the Research Network on Improving Australia's Data Mining and Knowledge Discovery Research (which received seeding support from the ARC), and the Institute of Analytic Professionals of Australia. It also strengthened the interconnections between researchers in academia and research organisations, and industry practitioners, who utilise data mining techniques in various business case studies, evidenced in the program of PAKDD2004 in Sydney.

Nowadays data mining efforts have gone beyond crunching databases of credit card usage or retail transaction records. As the data mining technologies are a core part of the so-called “embedded intelligence” in science, business, health care, drug design and other areas of human endeavour, consistent methodologies and reliable implementations are becoming critical to the success. This year **AusDM** continues to build on these trends, looking at the practice of data mining and business analytics. The conference looked specifically at working solutions and their real world implementations.

The conference received 43 submissions. We would like to thank all those who submitted their work to the conference. In an attempt to accommodate all valuable works, 21 submissions have been selected for publication and the presentation program has been extended to two days. **AusDM** follows a rigid blind peer-review process and ranking-based paper selection process. All papers were extensively reviewed by at least three referees drawn from the program committee. Accepted works have been grouped in five sessions and allocated equal presentation time slots.

Special thanks go to the program committee members and external reviewers, for the final quality of selected papers depends on their efforts. The **AusDM** review cycle runs on a very tight schedule and we would like to thank all reviewers for their commitment and professionalism.

Last but not least, we would like to thank the organisers of AI 2004 and Complex 2004 for hosting **AusDM**.

Simeon, J. Simoff and Graham J. Williams

December 2004

## Conference Chairs

Simeon J Simoff  
Graham J Williams

University of Technology, Sydney  
Australian Taxation Office, Canberra

## Program Committee

Mihael Ankerst	Boeing Corp., USA
Michael Bain	University of New South Wales
Rohan Baxter	Australian Taxation Office
Helmut Berger	University of Technology, Sydney, Australia
Michael Böhlen	Free University Bolzano-Bozen, Italy
Jie Chen	CSIRO, Canberra, Australia
Peter Christen	Australian National University, Australia
Thanh-Nghi Do	Can Tho University, Vietnam
Vladimir Estivill-Castro	Giffith University, Australia
Hongjian Fan	University of Melbourne, Australia
Eibe Frank	Waikato Univesity, New Zealand
Warwick Graco	Australian Taxation Office
Lifang Gu	CSIRO, Canberra, Australia
Tony Jan	University of Technology, Sydney, Australia
Warren Jin	CSIRO, Canberra, Australia
Paul Kennedy	University of Technology, Sydney, Australia
Inna Kolyshkina	Pricewaterhouse Coopers Actuarial, Sydney, Australia
Weiqiang Lin	Australian Taxation Office
John Maindonald	Australian National University
Mohamed Medhat Gaber	Monash University, Australia
Mark Norrie	Teradata, NCR
Robert Pearson	Health Insurance Commission, Australia
Tom Osborn	Wunderman, NUIX Pty Ltd, Australia
Francois Poulet	ESIEA-Pole ECD, Laval, France
Greg Saunders	University of Ballarat, Australia
David Skillicorn	Queen's University, Canada
John Yearwood	University of Ballarat, Australia
Osmar Zaiane	University of Alberta, Canada

## External reviewers

Catalina Maria-Luiza Antonie	University of Alberta, Canada
Hongxing He	CSIRO, Canberra, Australia

# AusDM04 Conference Program, 6<sup>th</sup> – 7<sup>th</sup> December 2004, Cairns, Australia

## Monday, 6 December, 2004

**9:20 - 9:30** Opening and Welcome

### **09:30 - 10:30 Session I: Trees**

- 09:30 - 10:00 ANALYSIS OF STRUCTURAL CONVERGENCE OF CONSOLIDATED TREES WHEN RESAMPLING IS REQUIRED  
Jesus M Perez, Javier Muguerza, **Olatz Arbelaitz**, Ibai Gurrutxaga, Jose I. Martin
- 10:00 - 10:30 K NEAREST NEIGHBOR EDITION TO GUIDE CLASSIFICATION TREE LEARNING  
**J. M. Martinez-Otzeta**, B. Sierra, E. Lazkano and A. Astigarraga

**10:30 - 11:00** Coffee break

### **11:00 - 12:30 Session II: Text Mining**

- 11:00 - 11:30 THE SCAMSEEK PROJECT TEXT MINING FOR FINANCIAL SCAMS ON THE INTERNET  
**Jon Patrick**
- 11:30 - 12:00 FUZZY DOCUMENT FILTER FOR THE INTERNET  
**Deepani B Guruge** and Russel J Stonier
- 12:00 - 12:30 INFORMING THE CURIOUS NEGOTIATOR: AUTOMATIC NEWS EXTRACTION FROM THE INTERNET  
**Debbie Zhang** and Simeon J. Simoff

**12:30 - 13:30** Lunch

### **13:30 - 15:00 Session III: Applications and Methodologies**

- 13:30 - 14:00 VISUALISATION AND EXPLORATION OF SCIENTIFIC DATA USING GRAPHS  
**Ben Raymond** and Lee Belbin
- 14:00 - 14:30 AN EVALUATION OF THE UTILITY OF TWO DATA MINING PROJECT METHODOLOGIES  
**Marcel van Rooyen**
- 14:30 - 15:00 DATA MINING APPLICATION IN A SOFTWARE PROJECT MANAGEMENT PROCESS  
**Richi Nayak** and Tian Qiu

**15:00 - 15:30** Coffee break

### **15:30 - 17:00 Session IV: Linking and Techniques**

- 15:30 - 16:00 A PROBABILISTIC GEOCODING SYSTEM BASED ON A NATIONAL ADDRESS FILE  
**Peter Christen**, Tim Churches and Alan Willmore
- 16:00 - 16:30 MINING OPTIMAL ITEM PACKAGES USING MIXED INTEGER PROGRAMMING  
N R Achuthan, Raj P. Gopalan and **Amit Rudra**
- 16:30 - 17:00 DECISION THEORETIC FUSION FRAMEWORK FOR ACTIONABILITY USING DATA MINING ON AN EMBEDDED SYSTEM  
**Heungkyu Lee** and Hanseok Ko

**Tuesday, 7 December, 2004**

**9:30 - 10:30 Session V: Health and Medical Data**

- 09:30 - 10:00 MINING MOUCLAS PATTERNS AND JUMPING MOUCLAS PATTERNS TO CONSTRUCT CLASSIFIERS  
Yalei Hao, Gerald Quirchmayr and **Markus Stumptner**
- 10:00 - 10:30 EXPLORATORY HEALTH DATA MINING: IDENTIFYING FACTORS ASSOCIATED WITH COLORECTAL CANCER  
Jie Chen, Hongxing He, Huidong Jin, Damien McAullay, **Graham Williams**, Chris Kelman

**10:30 - 11:00** Coffee break

**11:00 - 12:30 Session VI: Techniques and Sequence Mining**

- 11:00 - 11:30 EFFICIENTLY IDENTIFYING EXPLORATORY RULES' SIGNIFICANCE  
**Shiying Huang** and Geoffrey I. Webb
- 11:30 - 12:00 AN APPLICATION OF TIME-CHANGING FEATURE SELECTION  
**Yihao Zhang**, Mehmet A. Orgun, Weiqiang Lin and Warwick Graco
- 12:00 - 12:30 A MULTI-LEVEL FRAMEWORK FOR THE ANALYSIS OF SEQUENTIAL DATA  
**Carl H. Mooney**, Denise de Vries, John F. Roddick

## Table of Contents

Analysis of structural convergence of consolidated trees when resampling is required Jesus M Perez, Javier Muguerza, Olatz Arbelaitz, Ibai Gurrutxaga, Jose I. Martin .....	9
K Nearest neighbour edition to guide classification tree learning J. M. Martinez-Otzeta, B. Sierra, E. Lazkano and A. Astigarraga .....	23
The Scamseek – Project text mining for financial scams on the Internet Jon Patrick .....	33
Fuzzy document filter for the Internet Deepani B Guruge and Russel J Stonier .....	39
Informing the Curious Negotiator: Automatic news extraction from the Internet Debbie Zhang and Simeon J. Simoff .....	55
Visualisation and exploration of scientific data using graphs Ben Raymond and Lee Belbin .....	73
An evaluation of the utility of two data mining project methodologies Marcel van Rooyen .....	85
Data mining application in a software project management process Richi Nayak and Tian Qiu .....	99
A Probabilistic Geocoding System based on a National Address File Peter Christen, Tim Churches, and Alan Willmore .....	111
Mining optimal item packages using mixed integer programming N R Achuthan, Raj P. Gopalan and Amit Rudra .....	125
Decision theoretic fusion framework for actionability using data mining on an embedded system Heungkyu Lee and Hanseok Ko .....	137
Mining MOUCLAS patterns and jumping MOUCLAS patterns to construct classifiers Yalei Hao, Gerald Quirchmayr, Markus Stumptner .....	149
Exploratory health data mining: Identifying factors associated with colorectal cancer Jie Chen, Hongxing He, Huidong Jin, Damien McAullay, Graham Williams, Chris Kelman .....	157
Efficiently identifying exploratory rules' significance Shiyang Huang and Geoffrey I. Webb .....	169
An Application of Time-Changing Feature Selection Yihao Zhang, Mehmet A. Orgun, Weiqiang Lin and Warwick Graco .....	183
A multi-level framework for the analysis of sequential data Carl H. Mooney, Denise de Vries and John F. Roddick .....	199
Building a hierarchical hidden markov model: An application to health insurance data Ah Chung Tsoi, Shu Zhang, and Markus Hagenbuchner .....	215
A data mining approach to analyze the effect of cognitive style and subjective emotion on the accuracy of intuitive time-series forecasting Hung Kook Park, Byoungso Song, Hyeon-Joong Yoo, Dae Woong Rhee, Kang Ryoung Park and Juno Chang .....	231

Decision models for record linkage	
Lifang Gu and Rohan Baxter .....	241
Mining quantitative association rules in protein sequences	
Nitin Gupta, Nitin Mangal, Kamal Tiwari and Pabitra Mitra .....	255
Mining X-ray images of SARS patients	
Xie Xuanyang, Li Xi, Xu Yufeng, Wan Shouhong and Gong Yuchang .....	263
Author Index .....	281

# Analysis of structural convergence of Consolidated Trees when resampling is required

Jesús M. Pérez, Javier Muguerza, Olatz Arbelaitz, Ibai Gurrutxaga, José I. Martín

Dept. of Computer Architecture and Technology, University of the Basque Country  
M. Lardizabal, 1, 20018 Donostia, Spain  
{acppedej, jmuguerza, acpargao, acpgugoi, acpmaarj}@si.ehu.es  
<http://www.sc.ehu.es/aldapa>

**Abstract.** Resampling techniques are used in machine learning with different objectives: reducing the size of the training set, equilibrating the class imbalance or non-uniform cost error, etc. When different subsamples of the same data set are used to induce classification trees, the structure of the built classifiers is very different. The stability of the structure of the tree is of capital importance in many domains, such as illness diagnosis, fraud detection in different fields, customer's behaviour analysis (marketing), etc, where comprehensibility of the classifier is necessary. We have developed a methodology for building classification trees (Consolidated Trees) from multiple samples where the final classifier is a single decision tree. The paper presents an analysis of the structural stability of our algorithm versus C4.5 algorithm. The classification trees generated with our algorithm, achieve smaller error rates and structurally more steady trees than C4.5 when using resampling techniques. The main focus on this paper is showing how Consolidated Trees built with disjoint sets of subsamples tend to converge to the same tree when the number of used subsamples is increased.

## 1 Introduction

Many examples of the use of resampling techniques —oversampling or undersampling— with different objectives can be found in bibliography. A very important application of resampling is to use it in order to equilibrate the class distribution in databases with class imbalance [12],[18]. In many areas, such as medicine, fraud detection, etc; cases of one of the classes can be difficult to obtain. This leads very often to class imbalance in the data set which, in general, does not even coincide with the distribution expected in reality. A similar case is the one of databases with non-uniform cost, where the misclassification cost is not the same for the whole confusion matrix. In these cases, the use of resampling techniques to make some errors become more important than others can be a way of introducing such a cost in the learning algorithm, if the algorithm does not take into account the cost-matrix in the induction process [9]. On the other hand, for some databases the use of machine learning algorithms is computationally too expensive due to their memory requirements. In these cases resampling can be used for size reduction [4],[16]. We can not forget one of the most extended uses of resampling techniques: the

construction of multiple classifiers such as bagging, boosting, etc; able to obtain larger accuracy in the classification [1],[3],[6],[10].

In all the mentioned cases, subsamples obtained by resampling the original data set will be given to the learning algorithm in order to build a classifier. This resampling affects severely the behaviour of the classification algorithms [12]. Classification trees are not an exception. Classification trees induced from slightly different subsamples of the same data set are very different in accuracy and structure [8]. This weakness is called unsteadiness or instability. The stability is of capital importance in many domains, such as illness diagnosis, fraud detection in different fields, customer's behaviour analysis (marketing), etc, where comprehensibility of the classifier is necessary [7]. As Turney found when working on industrial applications of decision tree learning, "the engineers are disturbed when different batches of data from the same process result in radically different decision trees. The engineers lose confidence in the decision trees even when we can demonstrate that the trees have high predictive accuracy" [17]. Some authors [7],[17] have measured the stability of a classifier observing if different instances agree in the prediction made for each case of the test set (logical stability or variance). However, since the explanation of a tree comes from its structure we need a way of building structurally steady classifiers in order to obtain a convincing explanation (physical stability or structural stability).

This paper presents an analysis of the structural stability of decision trees built using the Consolidated Trees' Construction algorithm (CTC) when subsampling techniques are required (class imbalance, non-uniform cost, etc.). The CTC algorithm, opposite to other algorithms that work with many subsamples (bagging, boosting), induces a single tree, therefore it does not lose the comprehensibility of the base classifier. A measure of similarity between two induced concepts (tree's structures) will be used in order to evaluate the structural stability of the algorithm. The structural analysis done proves that the algorithm has a steadier behaviour than C4.5 [15], obtaining this way a steadier explanation. In this paper the main focus is done in showing how the trees built with the proposed algorithm tend to become more similar when the number of subsamples used to build them increases. In some domains, they converge to the same instance of tree even if the used subsamples are totally different.

The discriminating capacity of the CTC algorithm has already been evaluated in previous works [13],[14]. These works show that the classification trees generated using the new algorithm achieve smaller error rates than the ones built with C4.5, giving this way a better quality to the explanation.

The paper proceeds with a description of our methodology for building classification trees, Section 2. In Section 3, the description of the data set and the experimental set-up is presented. This paper includes a summary of the results of our previous work in Section 4. Section 5 presents the analysis of the structural stability and convergence of the structure of trees built with CTC algorithm. Finally, Section 6 is devoted to summarise the conclusions and further work.

## 2 Consolidated Trees' Construction algorithm

Consolidated Trees' Construction algorithm (CTC) uses several subsamples to build a single tree. This technique is radically different from bagging, boosting, etc. The

consensus is achieved at each step of the tree's building process and only one tree is built.

The different subsamples are used to make proposals about the feature that should be used to split in the current node. The split function used in this work is the gain ratio criterion (the same used by Quinlan in C4.5). The decision about which feature will be used to make the split in a node of the Consolidated Tree (CT) is accorded among the different proposals. The decision is made by a voting process node by node. Based on this decision, all the subsamples are divided using the same feature. The iterative process is described in Algorithm 1.

The algorithm starts extracting a set of subsamples from the original training set (*Number\_Samples*). The subsamples are obtained based on the desired resampling technique (*Resampling\_Mode*).

Decision tree's construction algorithms usually, divide the initial sample in several data partitions. In our algorithm,  $LS^i$  contains the data partitions created from each subsample  $S^i$ .

---

**Algorithm 1.** Consolidated Trees' Construction Algorithm (CTC)

---

Generate *Number\_Samples* subsamples ( $S^i$ ) from  $S$  with *Resampling\_Mode* method.

*CurrentNode* := *RootNode*

**for**  $i := 1$  to *Number\_Samples*

$LS^i := \{S^i\}$

**end for**

**repeat**

**for**  $i := 1$  to *Number\_Samples*

$CurrentS^i := First(LS^i)$

$LS^i := LS^i - CurrentS^i$

        Induce the best split  $(X, B)^i$  for  $CurrentS^i$

**end for**

    Obtain the consolidated pair  $(X_c, B_c)$ , based on  $(X, B)^i$ ,  $1 \leq i \leq$   
*Number\_Samples*

**if**  $(X_c, B_c) \neq Not\_Split$

        Split *CurrentNode* based on  $(X_c, B_c)$

**for**  $i := 1$  to *Number\_Samples*

            Divide  $CurrentS^i$  based on  $(X_c, B_c)$  to obtain  $n$  subsamples  $\{S_1^i, \dots, S_n^i\}$

$LS^i := \{S_1^i, \dots, S_n^i\} + LS^i$

**end for**

**else** consolidate *CurrentNode* as a leaf

**end if**

*CurrentNode* := *NextNode*

**until**  $\forall i, LS^i$  is empty

---

The pair  $(X, B)^i$  is the split proposal for the first partition in  $LS^i$ .  $X$  is the feature selected to split and  $B$  indicates the proposed branches or criteria to divide the data in the current node.  $X_c$  is the feature obtained by a voting process among all the proposed  $X$ . Whereas  $B_c$  is the median of the proposed *Cut* values when  $X_c$  is continuous and all the possible values of the feature when  $X_c$  is discrete.

When a node is consolidated as a leaf node, the a posteriori probabilities associated to it are calculated averaging the a posteriori obtained from the data partitions related to that node in all the subsamples.

The used resampling technique and the number of subsamples used in the tree's building process are important aspects of the algorithm [16]. There are many possible combinations for the *Resampling\_Mode*: size of the subsamples —100%, 75%, 50%, etc; of the original training set —, with replacement or without replacement, stratified or not, etc. The best results have been obtained with 75% without replacement and stratified, and these are the ones presented in the paper.

Once the consolidated tree has been built, it works the same way a decision tree does.

### 3 Experimental methodology

Twenty databases of real applications have been used for the experimentation. Most of them belong to the well known UCI Repository benchmark [2]. The Segment domain has been used for experimentation in two different ways: taking into account the whole set of data (*segment2310*) and conserving the training/test division of the original data set (*Segment210*). The *Faithful* database is a real data application from our environment, centred in the electrical appliance's sector. Table 1 shows the characteristics of the used domains.

**Table 1.** Description of experimental domains

<i>Domain</i>	<i>N. of patterns</i>	<i>N. of features</i>	<i>N. of classes</i>
<i>Breast-W</i>	699	10	2
<i>Heart-C</i>	303	13	2
<i>Hypo</i>	3163	25	2
<i>Lymph</i>	148	18	4
<i>Credit-G</i>	1000	20	2
<i>Segment210</i>	210	19	7
<i>Iris</i>	150	4	3
<i>Glass</i>	214	9	7
<i>Voting</i>	435	16	2
<i>Hepatitis</i>	155	19	2
<i>Soybean-L</i>	290	35	15
<i>Sick-E</i>	3163	25	2
<i>Liver</i>	345	6	2
<i>Credit-A</i>	690	14	2
<i>Vehicle</i>	846	18	4
<i>Breast-Y</i>	286	9	2
<i>Heart-H</i>	294	13	2
<i>Segment2310</i>	2310	19	7
<i>Spam</i>	4601	57	2
<i>Faithful</i>	24507	49	2

The CTC methodology has been compared to the C4.5 tree building algorithm Release 8 of Quinlan, using the default parameter settings. Both kinds of trees have been pruned, using the pruning algorithm of the C4.5 R8 software, to situate both

systems in a similar zone in the learning curve [11],[19]. We can not forget that developing too much a classification tree leads to a greater probability of overtraining. The methodology used for the experimentation is a 10-fold stratified cross validation [11]. In each of the folds of the cross-validation 100 stratified subsamples have been extracted, always without replacement and with size of 75% of the training sample in the corresponding fold. These subsamples have been used to build both kinds of trees, CT and C4.5.

For CTC algorithm the subsamples have been used disjointedly to build the trees, which has led to different number of instances of CTs when varying the parameter *Number\_Samples*: 5 (20 trees), 10 (10 trees), 20 (5 trees), 30 (3 trees), 40 (2 trees) and 50 (2 trees).

For C4.5 algorithm different options have been tried:

- C4.5<sub>100</sub> consists on building a tree with each one of the 100 subsamples mentioned before, generated undersampling the training set (fold). The amount of information of the original training set used by each algorithm is different in this case: a CT sees more information than a C4.5 tree, which can lead to a difference in accuracy. This has led us to design another comparison, where both algorithms use the same information (C4.5<sub>union</sub>).
- The sample used to induce each one of the C4.5<sub>union</sub> trees will be the union of the subsamples used to build the corresponding CT. So, in this experimentation the information handled by both algorithms is the same. In this case as many C4.5 trees as CTs are built.
- Related to the previous one we made a third comparison among C4.5 and CTC algorithm where the C4.5 trees have been built directly from the training data belonging to each fold of the 10-fold cross-validation (C4.5<sub>not resampling</sub>). We can not forget that the comparison presented in this case does not solve the problem described in the introduction of this article; the cases where resampling is compulsory due to different reasons. However we think the comparison is interesting to appreciate correctly the achieved error rates.

## 4 Summary of previous work

This section is devoted to present the results of different comparisons made among the two algorithms (C4.5 and CTC).

For better understanding the made comparison, we can not forget that the use of resampling techniques (oversampling or undersampling) is very extended in machine learning, for example in databases with class imbalance or non-uniform cost, and the objective of this work is the analysis of how resampling affects the compared algorithms.

The analysis has been made from two points of view: error and structural stability. In order to evaluate the structural stability a structural distance among the trees that are being compared has been defined: *Common*. This structural measure is based on a pair to pair comparison, *Similarity*, among all the trees of the set. This function counts the common nodes among two trees. If two nodes will be counted as common nodes, they have to coincide in the feature used to make the split, the proposed branches or

stratification and the position in the tree. This value is calculated starting from the root and covering the tree, level by level. When a different node is found the subtree under that node is not taken into account. For a set of trees  $T_{set}$ , with  $m$  trees the *Common* value is calculated as the average value of all the possible pair to pair comparisons:

$$Common(T_{set}) = \frac{2}{m(m-1)} \sum_{\substack{k,l=0 \\ k < l}}^{m-1} Similarity(T_k, T_l)$$

From a practical point of view, *Common* quantifies structural stability of the classification algorithm, whereas the error would quantify the quality of the explanation given by the tree. Evidently it is not enough to have a greater value in *Common* to have greater explaining capacity; it is compulsory to have a similar or smaller error rate. This has been our main goal.

As a summary of previous work we can say that the behaviour of the CTC algorithm improves when the value of *Number\_Samples* increases. When this value is 20 or greater, the results for CTC are better in average than results for any of the versions of C4.5. Table 2 shows the results of the comparison of CTC (with *Number\_Samples* = 30), C4.5<sub>100</sub>, C4.5<sub>union</sub>, and C4.5<sub>not\_resampling</sub>.

**Table 2.** Results of Error and *Common* for every domain. CTC (*Number\_Samples* = 30), C4.5<sub>100</sub> (C4.5<sub>l</sub>), C4.5<sub>union</sub> (C4.5<sub>u</sub>) and C4.5<sub>not\_resampling</sub> (C4.5<sub>n\_r</sub>) are shown.

	Error								Common			
	CTC	C4.5 <sub>l</sub>	R.Dif	C4.5 <sub>u</sub>	R.Dif	C4.5 <sub>n_r</sub>	R.Dif	CTC	C4.5 <sub>l</sub>	C4.5 <sub>u</sub>	C4.5 <sub>n_r</sub>	
<i>Breast-W</i>	5,89	6,30	-0,06	7,10	-0,17	6,03	-0,02	2,90	1,71	19,17	1,93	
<i>Heart-C</i>	22,33	23,92	-0,07 <sup>†</sup>	27,38	-0,18 <sup>†</sup>	23,42	-0,05	6,87	1,43	16,30	3,24	
<i>Hypo</i>	0,76	0,79	-0,04	1,26	-0,40 <sup>†</sup>	0,73	0,04	4,17	2,67	24,30	3,49	
<i>Lymph</i>	18,27	21,12	-0,14 <sup>†</sup>	25,00	-0,27 <sup>†</sup>	18,06	0,01	8,90	2,22	18,50	3,62	
<i>Credit-G</i>	26,97	27,75	-0,03	31,90	-0,15 <sup>†</sup>	27,80	-0,03	14,00	2,34	40,97	4,87	
<i>Segment210</i>	10,80	11,98	-0,10 <sup>†</sup>	10,41	0,04	10,88	-0,01	5,03	2,09	10,43	3,00	
<i>Iris</i>	4,23	6,38	-0,34 <sup>†</sup>	6,02	-0,30	5,35	-0,26	2,67	2,05	5,53	3,46	
<i>Glass</i>	29,18	32,04	-0,09	29,00	0,01	29,25	0,00	7,37	2,65	15,60	7,40	
<i>Voting</i>	3,36	4,18	-0,20 <sup>†</sup>	4,59	-0,27	3,44	-0,02	4,60	2,19	22,13	3,75	
<i>Hepatitis</i>	19,30	20,45	-0,06	20,92	-0,08	18,66	0,03	3,17	0,84	11,93	3,06	
<i>Soybean-L</i>	9,97	13,10	-0,24 <sup>†</sup>	11,36	-0,12	9,29	0,07	14,57	6,27	22,87	10,24	
<i>Sick-E</i>	2,34	2,18	0,07	2,95	-0,21	1,93	0,18	8,90	4,77	17,83	9,09	
<i>Liver</i>	34,50	36,11	-0,04	36,95	-0,07	37,93	-0,10	9,27	1,18	18,33	2,38	
<i>Credit-A</i>	15,20	14,86	0,02	18,19	-0,16 <sup>†</sup>	15,2	0,00	5,77	2,11	27,20	3,13	
<i>Vehicle</i>	27,40	28,58	-0,04	26,93	0,02	28,49	-0,04	16,87	7,13	30,00	14,28	
<i>Breast-Y</i>	26,36	28,11	-0,06	35,50	-0,26 <sup>†</sup>	25,19	0,04	2,20	0,71	33,83	1,60	
<i>Heart-H</i>	21,96	21,61	0,02	22,85	-0,04	22,13	-0,01	5,27	1,37	25,03	1,58	
<i>Segment2310</i>	3,23	3,96	-0,18 <sup>†</sup>	3,23	0,00	3,51	-0,09	22,20	10,39	26,13	14,51	
<i>Spam</i>	7,29	7,68	-0,05 <sup>†</sup>	7,77	-0,06	6,84	0,06	13,87	4,50	30,90	9,93	
<i>Faithful</i>	1,47	1,52	-0,03 <sup>†</sup>	2,35	-0,37 <sup>†</sup>	1,52	-0,03	11,90	6,58	57,13	7,78	
Average 75%	14,54	15,63	-0,08	16,58	-0,15	14,78	-0,01	8,52	3,26	23,71	5,62	

Values related to Error and *Common* are given (column R.Dif is always calculated as the relative difference among the CTC results and the results of C4.5). The table shows that in 17 (C4.5<sub>100</sub>), 15 (C4.5<sub>union</sub>) and 11 (C4.5<sub>not\_resampling</sub>) domains out of 20, the error is smaller for CTC than for C4.5. The statistically significant differences (the paired t-test [5],[6]), with a confidence level of 95%, have been marked with <sup>†</sup>. The differences are statistically significant in 9 databases for C4.5<sub>100</sub>, and 7 databases for

C4.5<sub>union</sub>. In the databases where results for C4.5<sub>100</sub> or C4.5<sub>union</sub> are better, the differences are not statistically significant. The differences with results of C4.5<sub>not\_resampling</sub> are never statistically significant being the behaviour of CTC better in average. So we can ensure that the discriminating capacity of CTC algorithm is at least as good or better than the one of C4.5. In this situation, it is worth the comparison of the structural stability of the different classifiers. Achieving greater structural stability will mean that CT trees have better explaining capacity. The data show that CTs achieve a higher structural stability than C4.5<sub>100</sub> (in average 8,52 compared to 3,26) and C4.5<sub>not\_resampling</sub> (in average 8,52 compared to 5,62). There is an exception in *Faithful* database. This happens because the complexity of C4.5 trees built for this database, is an order of magnitude larger, but, error is smaller for CTC and the difference is statistically significant.

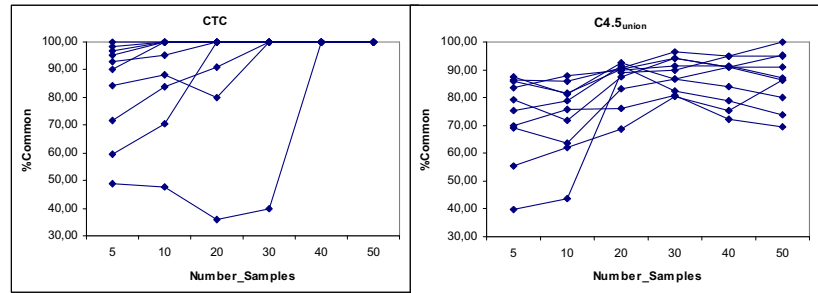
Looking to the values of *Common* obtained for C4.5<sub>union</sub> we could say that they achieve higher structural stability than CTC (*Common* is in average 23,71 compared to 8,52) but this happens because complexity of C4.5<sub>union</sub> trees is an order of magnitude larger than the complexity of CTs. So, being the error smaller for CTC, the principle of parsimony of the model makes worse the C4.5<sub>union</sub>. More information about this experimentation can be found in [14].

Therefore, we can say that in average, classification trees induced with the CTC algorithm have a lower error rate than those induced with C4.5, and they are structurally steadier. As a consequence they provide a wider and steadier explanation when resampling is necessary, that allows to deal with the problem of the excessive sensitivity classification trees have to resampling methods.

## 5 Analysis of convergence

We have observed that the value of *Common* for CT trees increases with the number of used subsamples. This means that the CT trees tend to have a larger common structure when *Number\_Samples* increases. This is a desirable behaviour but it could be due to the higher complexity of the trees (this was the case of C4.5<sub>union</sub> in previous section). In order to take into account the parsimony principle we have normalised the *Common* value in respect to the trees' size (number of internal nodes). We will denominate this measure *%Common*.

Fig. 1 shows the structural convergence for *Breast-W* domain (CTs, left side and C4.5<sub>union</sub> right side). The curves represent the values of *%Common* in each one of the folds of the cross-validation when the *Number\_Samples* parameter varies. We will give some clues for better understanding the figure. Obtaining a value of 100% for *%Common* in a set of trees means that all the compared trees are equal; obtaining a value of 90% means that in average the compared trees have 90% of the structure identical.



**Fig. 1.** Structural convergence of the CTC and the  $C4.5_{union}$  for the *Breast-W* domain.

Fig. 1 shows the evolution of  $\%Common$  for CTC algorithm (left side), when the number of samples used to build the trees increases, in each one of the 10 folds (each line represents one fold). For  $Number\_Samples = 5$ , 20 trees are compared in each fold. It can be observed that the CTs have in average 90% or more of the structure common in 6 folds out of 10; and in the fold with worse results the compared trees have 50% of the structure equal. As the number of samples used to build the CTs increases, the percentage of the trees that is equal increases in most of the folds. Concretely, when the number of samples used is 40 or greater, all the trees are identical in the 10 folds. We can say in this case that the CT trees converge structurally in  $Number\_Samples = 40$ .

For  $C4.5_{union}$  trees (right side), we can not observe any convergence when increasing the number of samples.

As a summary, we can say that for *Breast-W* database, CTs converge to an unique tree after a certain value of  $Number\_Samples$ , whereas  $C4.5$  trees show a greater structural variation.

**Table 3.** Results of  $\%Common$  for every domain.  $C4.5_{100}$ , CTC and  $C4.5_{union}$ .

%Common	$C4.5_{100}$	CTC						$C4.5_{union}$					
		5	10	20	30	40	50	5	10	20	30	40	50
<i>Breast-W</i>	61	84	89	91	94	100	100	73	73	86	88	86	86
<i>Heart-C</i>	13	24	35	47	54	63	59	13	20	30	35	44	43
<i>Hypo</i>	56	73	77	83	95	98	99	46	55	57	56	57	58
<i>Lymph</i>	32	64	78	90	94	94	94	63	68	76	80	77	82
<i>Credit-G</i>	8	14	18	28	36	28	44	11	15	21	25	28	29
<i>Segment210</i>	20	27	35	41	37	37	43	27	32	41	56	60	42
<i>Iris</i>	70	81	85	85	85	78	91	64	68	78	70	75	83
<i>Glass</i>	15	21	24	29	31	31	31	21	30	35	41	27	39
<i>Voting</i>	54	76	93	98	94	92	98	52	59	67	69	70	69
<i>Hepatitis</i>	16	33	44	49	46	70	52	19	26	35	38	48	43
<i>Soybean-L</i>	32	45	51	62	65	58	60	52	57	62	67	79	71
<i>Sick-E</i>	46	53	56	63	68	69	70	17	16	16	19	21	24
<i>Liver</i>	7	9	13	20	27	27	28	7	11	20	26	33	30
<i>Credit-A</i>	23	28	33	39	47	53	62	18	24	28	36	41	45
<i>Vehicle</i>	14	14	17	20	19	24	19	17	20	23	23	32	35
<i>Breast-Y</i>	19	39	45	54	68	87	77	27	43	51	55	59	59
<i>Heart-H</i>	23	31	38	40	62	62	65	30	27	37	35	44	45
<i>Segment2310</i>	29	36	37	42	47	45	51	33	38	46	42	49	56
<i>Spam</i>	5	9	10	14	13	12	18	6	9	11	12	12	11
<i>Faithful</i>	18	30	39	33	39	35	18	7	7	9	11	15	7
Average	28	40	45	51	56	58	59	30	35	41	44	48	48

If we analyse the results of the 20 databases (see Table 3 where averages of the 10 folds for  $\%Common$  are presented), for most of them (15 databases for  $Number\_Samples = 50$ , and similar values for the rest) the behaviour of CTC is better than the behaviour of  $C4.5_{union}$ . For some values of  $Number\_Samples$  parameter, relative improvements up to 50% are achieved.

After this study, it seems that from a certain value of  $Number\_Samples$  parameter the tree obtained with CTC algorithm will be always the same.

The previous analysis has been done comparing only trees with the same value of  $Number\_Samples$  parameter and we have observed that the value of  $\%Common$  increases with this parameter. This suggests us a new question: will also the structure of CTs built with different values of  $Number\_Samples$  be similar? In this case, we could say that CT trees are gradually changing towards a specific tree while  $Number\_Samples$  increases. To answer this question we present the study of Fig. 2.

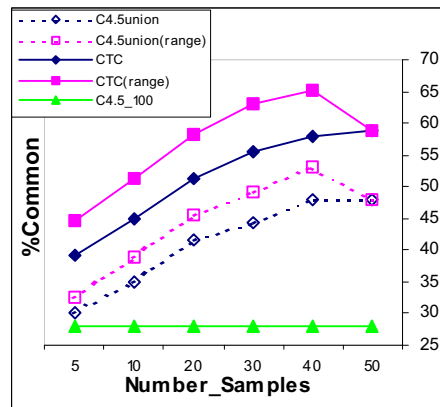


Fig. 2. Mean of  $\%Common$  for CTC,  $C4.5_{100}$  and  $C4.5_{union}$ .

Fig. 2 shows the values  $\%Common$  for CTC (continuous lines), and  $C4.5_{100}$  (triangles,  $Number\_Samples$  parameter does not make any sense in this case). So that, for each case an idea of the percentage of the tree that remains common is given. For each database, average  $\%Common$  values of the 10 folds are calculated and every point in the graphic represents the average of the 20 databases. For CTC and  $C4.5_{union}$  two studies are presented. In the first one, trees built with the same  $Number\_Samples$  value are compared (diamonds). In the second one, “range” in Fig. 2, trees with different values of  $Number\_Samples$  are compared (squares). In this case the point corresponding to  $Number\_Samples = 20$  represents the  $\%Common$  value obtained from the comparison of every tree built with  $Number\_Samples \geq 20$ . Being this value (58) larger than  $\%Common$  (51) means that there are trees built with 30, 40 or 50 subsamples that when compared to the trees built with 20 subsamples all together, they are even more similar than the trees built with 20 subsamples among them.

On the other hand, it can be observed that the trees built using CTC have a larger common structure than the rest. In average we can say that for any value of  $Number\_Samples$ , CTC results are better in at least 10%. In the case of  $C4.5_{100}$ , the behaviour is much worse. Besides, being the values of CTC(range) larger than values of CTC, we can assert that independently of the used  $Number\_Samples$ , similar

structures are reached, so, we can say that even if different subsamples are used to build trees, the obtained structures are similar. This makes the explanation of the classification steady when varying the *Number\_Samples* parameter. It seems that for *Breast\_W* database, when *Number\_Samples* is greater than 40 all the trees are identical, and looking to the tendencies of the average, we could think that it will exist for each database a value of *Number\_Samples* with the same properties.

The data in Table 3 has given us the idea of studying the number of folds (*#folds*) where all the trees converge exactly to the same tree for the different values of *Number\_Samples*. Centring the analysis in CTC, we can differentiate three kinds of behaviours (clusters) among the analysed databases: domains where for the majority of folds (*#folds*  $\geq 7$ ) all the trees converge to the same one (Cluster1: *Breast-W*, *Hypo*, *Lymph*, *Iris*, *Voting*, *Breast-Y*), domains with an intermediate number of folds that converge to the same tree (Cluster2: *Heart-C*, *Hepatitis*, *Soybean-L*, *Heart-H*) and domains where for the analysed values of *Number\_Samples* this situation never happens (Cluster3: *Credit-G*, *Segment210*, *Glass*, *Sick-E*, *Liver*, *Credit-A*, *Vehicle*, *Segment2310*, *Spam*, *Faithful*).

Table 4 shows the results of the mentioned analysis for CTC and C4.5<sub>union</sub>.

**Table 4.** Analysis of converging folds (*#folds*) and %Common (%Com) for CTC and C4.5<sub>union</sub> for different Number\_Samples (*N\_S*)

		CTC						C4.5 <sub>union</sub>					
<i>N_S</i>		5	10	20	30	40	50	5	10	20	30	40	50
#folds	Cluster1	0,17	2,17	4,67	6,50	7,83	8,33	0,00	0,00	0,17	0,33	0,67	1,00
	Cluster2	0,00	0,00	0,00	0,50	1,75	1,25	0,00	0,00	0,00	0,00	0,00	0,25
	Cluster3	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,10	0,00
%Com	Cluster1	69,7	77,7	83,5	88,4	91,3	93,0	54,3	61,2	69,2	69,7	70,8	72,9
	Cluster2	33,0	41,8	49,7	56,5	63,3	59,3	28,6	32,5	41,1	43,9	53,9	50,8
	Cluster3	24,1	27,2	32,9	36,4	36,2	38,4	16,4	20,2	25,0	29,1	31,7	31,7

When trying to understand the values in Table 4, it has to be taken into account that we use very hard conditions to count an unity: all the trees built for that value of *Number\_Samples* have to be identical. For example, in *Breast\_W* and *Number\_Samples* = 20 the values of %Common in the different folds are: 35,71; 80,00; 90,91; and for the remaining seven 100,00 (see Fig.1). This means that in seven folds all the compared trees are identical. In this case the value of *#folds* would be 7. The values shown in the table are averages of the databases belonging to the corresponding cluster.

Table 4 shows that the number of converging folds increases with the parameter *Number\_Samples* for both algorithms. On the other hand, values obtained for CTC are always much better than values for C4.5<sub>union</sub> in the 3 analysed sets. Besides, in every database the error of the CT trees is smaller than error of C4.5<sub>union</sub> or C4.5<sub>100</sub> trees and, as it can be observed in Table 2, most of the domains in Cluster1 are among the databases where the differences are statistically significant.

The same kind of analysis has been done for trees built with C4. 5<sub>100</sub> option. The number of folds where all the trees converge to the same one in this case is 0 for every database. The percentage of average common structure (%Common) is 28%

(See Table 3); even lower than the values obtained for CT trees with *Number\_Samples*=5 (40%).

Therefore the CTC algorithm provides a wider and steadier explanation with smaller error rates.

## 6 Conclusions and further work

In order to afford the unsteadiness classification trees suffer when small changes in the training set happen, we have developed a methodology for building classification trees: Consolidated Trees' Construction Algorithm (CTC), being the objective to maintain the explanation without losing accuracy. This paper focuses on the study of the structural convergence of the algorithm.

The behaviour of the CTC algorithm has been compared to C4.5 for twenty databases of the UCI Repository.

The results show that CT trees tend to converge to a single tree when *Number\_Samples* is increased and the obtained classification trees achieve besides, smaller error rates than C4.5. So we can say that this methodology builds structurally more steady trees, giving stability to the explanation and with smaller error rate, so, with higher quality in the explanation. This is essential for some specific domains: medical diagnosis, fraud detection, etc.

Observing the results in structural stability we can conclude that the number of samples required to achieve the structural convergence changes depending on the database. We are analysing the convergence for larger values of the parameter *Number\_Samples* in order to find the needed number of samples to achieve the convergence in each database. In this sense, the use of different parallelisation techniques (shared memory and distributed memory computers) will be considered due to the increase of computational cost.

Analysis of the results obtained for both algorithms with other percentage values for the *Resampling\_Mode* parameter can also be interesting.

The reasons that lead to three different clusters of domains in convergence need to be analysed. The analysis of the influence of the pruning in the error (bias/variance) can be interesting in this study.

The CTC algorithm provides a way to deal with the need of resampling the training set, either for a class imbalance problem or a size problem. Anyway, we are working in quantifying the influence that changes in the class distribution can have in the CTC algorithm. It would also be interesting the comparison of the results obtained with other techniques that use resampling in order to improve the accuracy of the classifier, such as bagging, boosting, etc., although they miss completely the explaining capacity.

## Acknowledgments

The work described in this paper was partly done under the University of Basque Country (UPV/EHU) project: 1/UPV 00139.226-T-14882/2002. It was also funded by the Diputación Foral de Guipuzcoa and the European Union.

We would like to thank the company Fagor Electrodomeísticos, S. COOP. for permitting us the use of their data (*Faithful*) obtained through the project BETIKO. The *lymphography* domain was obtained from the University Medical Centre, Institute of Oncology, Ljubljana, Yugoslavia. Thanks go to M. Zwitter and M. Soklic for providing the data.

## References

1. Bauer E., Kohavi R.: An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants, *Machine Learning*, Vol. 36, (1999) 105-139.
2. Blake, C.L., Merz, C.J.: UCI Repository of Machine Learning Databases, University of California, Irvine, Dept. of Information and Computer Sciences. <http://www.ics.uci.edu/~mlearn/MLRepository.html> (1998).
3. Breiman L.: Bagging Predictors. *Machine Learning*, Vol. 24, (1996) 123-140.
4. Chan P.K., Stolfo S.J.: Toward Scalable Learning with Non-uniform Class and Cost Distributions: A Case Study in Credit Card Fraud Detection, *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, (1998) 164-168.
5. Dietterich T.G.: Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms, *Neural Computation*, Vol. 10, No. 7, (1998) 1895-1924.
6. Dietterich T.G.: An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization, *Machine Learning*, Vol. 40, (2000) 139-157.
7. Domingos P.: Knowledge acquisition from examples via multiple models. *Proc. 14<sup>th</sup> International Conference on Machine Learning* Nashville, TN (1997) 98-106.
8. Drummond C., Holte R.C.: Exploiting the Cost (In)sensitivity of Decision Tree Splitting Criteria, *Proceedings of the 17th International Conference on Machine Learning*, (2000) 239-246.
9. Elkan C.: The Foundations of Cost-Sensitive Learning, *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, (2001) 973-978.
10. Freund, Y., Schapire, R. E.: Experiments with a New Boosting Algorithm, *Proceedings of the 13th International Conference on Machine Learning*, (1996) 148-156.
11. Hastie T., Tibshirani R., Friedman J.: *The Elements of Statistical Learning*. Springer-Verlag (es). ISBN: 0-387-95284-5, (2001).
12. Japkowicz N.: Learning from Imbalanced Data Sets: A Comparison of Various Strategies, *Proceedings of the AAAI Workshop on Learning from Imbalanced Data Sets*, Menlo Park, CA, (2000).
13. Pérez J.M., Muguerza J., Arbelaitz O., Gurrutxaga I.: A new algorithm to build consolidated trees: study of the error rate and steadiness, *Proceedings of the conference on Intelligent Information Systems*, Zakopane, Poland, (2004).
14. Pérez J.M., Muguerza J., Arbelaitz O., Gurrutxaga I., Martín J.I.: Behaviour of Consolidated Trees when using Resampling Techniques, *Proceedings of the 4th International Workshop on Pattern Recognition in Information Systems*, PRIS, Porto, Portugal, (2004).

15. Quinlan J.R.: C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers Inc.(eds), San Mateo, California (1993).
16. Skurichina M., Kuncheva L.I., Duin R.P.W. Bagging and Boosting for the Nearest Mean Classifier: Effects of Sample Size on Diversity and Accuracy, LNCS Vol. 2364. Multiple Classifier Systems: Proc. 3th Inter. Workshop, MCS , Cagliari, Italy, (2002) 62-71.
17. Turney P. Bias and the quantification of stability. Machine Learning, 20 (1995), 23-33.
18. Weiss G.M., Provost F.: Learning when Training Data are Costly: The Effect of Class Distribution on Tree Induction, Journal of Artificial Intelligence Research, Vol. 19, (2003) 315-354.
19. Windeatt T., Ardeshtir G.: Boosted Tree Ensembles for Solving Multiclass Problems, LNCS Vol. 2364. Multiple Classifier Systems: Proc. 3th Inter. Workshop, MCS , Cagliari, Italy, (2002) 42-51.



# K Nearest Neighbor Edition to Guide Classification Tree Learning

J. M. Martínez-Otzeta, B. Sierra, E. Lazkano and A. Astigarraga

Department of Computer Science and Artificial Intelligence  
University of the Basque Country  
P. Manuel Lardizabal 1, 20018 Donostia-San Sebastián  
Basque Country, Spain. e-mail: ccbmaotj@si.ehu.es  
<http://www.sc.ehu.es/ccwrobot>

**Abstract.** This paper presents a new hybrid classifier that combines the Nearest Neighbor distance based algorithm with the Classification Tree paradigm. The Nearest Neighbor algorithm is used as a preprocessing algorithm in order to obtain a modified training database for the posterior learning of the classification tree structure; experimental section shows the results obtained by the new algorithm; comparing these results with those obtained by the classification trees when induced from the original training data we obtain that the new approach performs better or equal according to the Wilcoxon signed rank statistical test.

**Keywords** Machine Learning, Supervised Classification, Classifier Combination, Classification Trees.

## 1 Introduction

Classifier Combination is an extended terminology used in the Machine Learning [19], more specifically in the *Supervised Pattern Recognition* area, to point out the supervised classification approaches in which several classifiers are brought to contribute to the same task of recognition [6]. Combining the predictions of a set of component classifiers has been shown to yield accuracy higher than the most accurate component on a long variety of supervised classification problems. To do the combinations, various strategies of decisions, implying these classifiers in different ways are possible [32] [14] [6] [27]. Good introductions to the area can be found in [8] and [9].

Classifier combination can fuse together different information sources to utilize their complementary information. The sources can be multi-modal, such as speech and vision, but can also be transformations [13] or partitions [4] [20] [22] of the same signal.

The combination, mixture, or ensemble of classification models could be performed mainly by means of two approaches:

- Concurrent execution of some paradigms with a posterior combination of the individual decision each model has given to the case to classify [31]. The

combination can be done by a voting approach or by means of more complex approaches [10].

- Hybrid approaches, in which the foundations of two or more different classification systems are implemented together in one classifier [13]. In the hybrid approach lies the concept of reductionism, where complex problems are solved through stepwise decomposition [28].

In this paper, we present a new hybrid classifier based on two families of well known classification methods; the first one is a distance based classifier [5] and the second one is the classification tree paradigm [2] which is combined with the former in the classification process. The  $k$ -NN algorithm is used as a preprocessing algorithm in order to obtain a modified training database for the posterior learning of the classification tree structure. We show the results obtained by the new approach and compare it with the results obtained by the classification tree induction algorithm (ID3 [23]).

The rest of the paper is organized as follows. Section 2 reviews the decision tree paradigm, while section 3 presents the K-NN method. The new proposed approach is presented in section 4 and results obtained are shown in section 5. Final section is dedicated to conclusions and points out the future work.

## 2 Decision Trees

A *decision tree* consists of nodes and branches to partition a set of samples into a set of covering decision rules. In each node, a single test or decision is made to obtain a partition. The starting node is usually referred as the root node. An illustration of this appears in Figure 1. In the terminal nodes or leaves a decision is made on the class assignment. Figure 2 shows an illustrative example of a Classification Tree obtained by the mineset software from SGI.

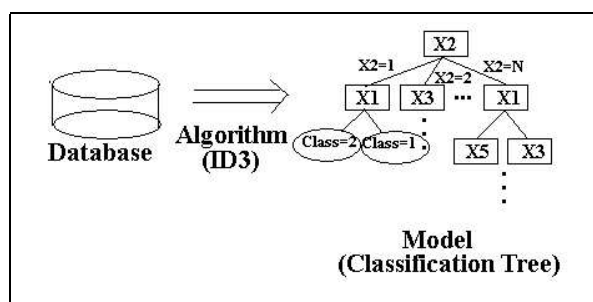
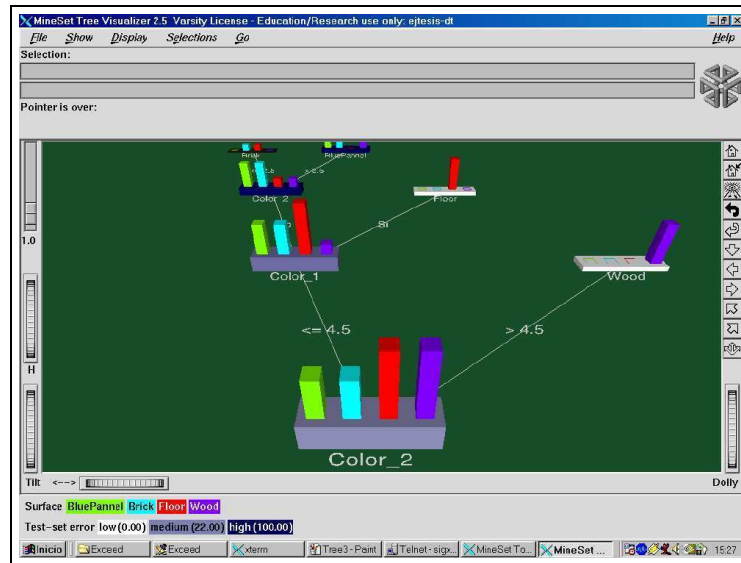


Fig. 1. Single classifier construction. Induction of a Classification Tree



**Fig. 2.** Example of a Classification Tree.

In each node, the main task is to select an attribute that makes the best partition between the classes of the samples in the training set. There are many different measures to select the best attribute in a node of the decision trees: two works gathering these measures are [18] and [15]. In more complex works like [21] these tests are made applying the linear discriminant approach in each node. In the induction of a decision tree, an usual problem is the overfitting of the tree to the training dataset, producing an excessive expansion of the tree and consequently losing predictive accuracy to classify new unseen cases. This problem is overcome in two ways:

- weighing the discriminant capability of the attribute selected, and thus discarding a possible successive splitting of the dataset. This technique is known as "prepruning".
- after allowing a huge expansion of the tree, we could revise a splitting mode in a node removing branches and leaves, and only maintaining the node. This technique is known as "postpruning".

The works that have inspired a lot of successive papers in the task of the decision trees are [2] and [23]. In our experiments, we use the well-known decision tree induction algorithm, ID3 [23].

### 3 The $K$ -NN Classification Method

A set of pairs  $(x_1, \theta_1), (x_2, \theta_2), \dots, (x_n, \theta_n)$  is given, where the  $x_i$ 's take values in a metric space  $X$  upon which is defined a metric  $d$  and the  $\theta_i$ 's take values in the set  $\{1, 2, \dots, M\}$  of possible classes. Each  $\theta_i$  is considered to be the index of the category to which the  $i$ th individual belongs, and each  $x_i$  is the outcome of the set of measurements made upon that individual. We use to say that " $x_i$  belongs to  $\theta_i$ " when we mean precisely that the  $i$ th individual, upon which measurements  $x_i$  have been observed, belongs to category  $\theta_i$ .

A new pair  $(x, \theta)$  is given, where only the measurement  $x$  is observable, and it is desired to estimate  $\theta$  by using the information contained in the set of correctly classified points. We shall call

$$x'_n \in x_1, x_2, \dots, x_n$$

the nearest neighbor of  $x$  if

$$\min d(x_i, x) = d(x'_n, x) \quad i = 1, 2, \dots, n$$

The NN classification decision method gives to  $x$  the category  $\theta'_n$  of its nearest neighbor  $x'_n$ . In case of tie for the nearest neighbor, the decision rule has to be modified in order to break it. A mistake is made if  $\theta'_n \neq \theta$ .

An immediate extension to this decision rule is the so called  $k$ -NN approach [3], which assigns to the candidate  $x$  the class which is most frequently represented in the  $k$  nearest neighbors to  $x$ . In Figure 3, for example, the 3-NN decision rule would decide  $x$  as belonging to class  $\theta_o$  because two of the three nearest neighbors of  $x$  belongs to class  $\theta_o$ .

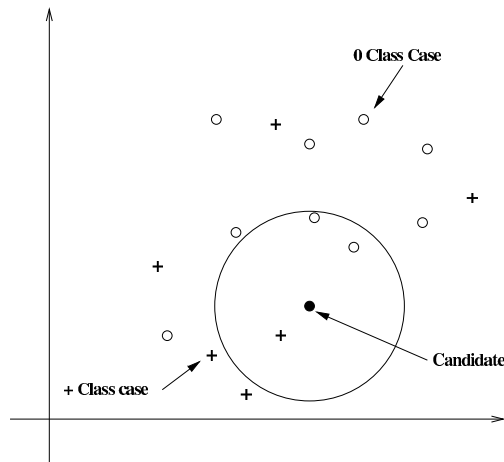
Much research has been devoted to the  $K$ -NN rule [5]. One of the most important results is that  $K$ -NN has asymptotically very good performance. Loosely speaking, for a very large design set, the expected probability of incorrect classifications (error)  $R$  achievable with  $K$ -NN is bounded as follows:

$$R^* < R < 2R^*$$

where  $R^*$  is the optimal (minimal) error rate for the underlying distributions  $p_i, i = 1, 2, \dots, M$ .

This performance, however, is demonstrated for the training set size tending to infinity, and thus, is not really applicable to real world problems, in which we usually have a training set of about hundreds or thousands cases, too little, anyway, for the number of probability estimations to be done.

More extensions to the  $k$ -NN approach could be seen in [5] [1] [25] [16]. More effort has to be done in the  $K$ -NN paradigm in order to reduce the number of cases of the training database to obtain faster classifications [5] [26].



**Fig. 3.** 3-NN classification method. A voting method has to be implemented to take the final decision. The classification given in this example by simple voting would be class=circle.

## 4 Proposed Approach

In boosting techniques, a distribution or set of weights over the training set is maintained. On each execution, the weights of incorrectly classified examples are increased so that the base learner is forced to focus on the hard examples in the training set. A good description of boosting can be found in [7].

Following the idea of focusing in the hard examples, we wanted to know if one algorithm could be used to boost a different one, in a simple way. We have chosen two well-known algorithms,  $k$ -NN and ID3, and our approach (in the following we will refer to it as  $k$ -NN-boosting) works as follows:

- Find the incorrectly classified instances in the training set using  $k$ -NN over the training set but the instance to be classified
- Duplicate the instances incorrectly classified in the previous step
- Apply ID3 to the augmented training set

Let us note that this approach is equivalent to duplicate the weight of incorrectly classified instances, according to  $k$ -NN.

In this manner, the core of this new approach consists of inflating the training database adding the cases misclassified by the  $k$ -NN algorithm, and then learn the classification tree from the new database obtained. It has to be said that this approach increases the computational cost only in the model induction phase, while the classification costs are the same as in the original ID3 paradigm.

## 5 Experimental Results

Ten databases are used to test our hypothesis. All of them are obtained from the *UCI Machine Learning Repository* [20]. These domains are public at the Statlog project WEB page [17]. The characteristics of the databases are given in Table 1. As it can be seen, we have chosen different types of databases, selecting some of them with a large number of predictor variables, or with a large number of cases and some multi-class problems.

**Table 1.** Details of databases

<i>Database</i>	<i>Number of cases</i>	<i>Number of classes</i>	<i>Number of attributes</i>
Diabetes	768	2	8
Australian	690	2	14
Heart	270	2	13
Monk2	432	2	6
Wine	178	3	13
Zoo	101	7	16
Waveform-21	5000	3	21
Nettalk	14471	324	203
Letter	20000	26	16
Shuttle	58000	7	9

In order to give a real perspective of applied methods, we use 10-Fold Cross-validation [29] in all experiments. All databases have been randomly separated into ten sets of training data and its corresponding test data. Obviously all the validation files used have been always the same for the two algorithms: ID3 and our approach,  $k$ -NN-boosting. Ten executions for every 10-fold set have been carried out using  $k$ -NN-boosting, one for each different K ranging from 1 to 10. In Table 2 a comparative of ID3 error rate, as well as the best and worst performance of  $k$ -NN-boosting, along with the average error rate among the ten first values of K, used in the experiment, is shown. The cases when  $k$ -NN-boosting outperforms ID3 are drawn in boldface. Let us note that in six out of ten databases the average of the ten sets of executions of  $k$ -NN-boosting outperforms ID3 and in two of the remaining four cases the performance is similar.

In nine out of ten databases there exists a value of K for which  $k$ -NN-boosting outperforms ID3. In the remaining case the performance is similar. In two out of ten databases even in the case of the worst K value with respect to accuracy,  $k$ -NN-boosting outperforms ID3, and in other three they behave in a similar way. In Table 3 the results of applying the Wilcoxon signed rank test [30] to compare the relative performance of ID3 and  $k$ -NN-boosting for the ten databases tested are shown.

**Table 2.** Rates of experimental errors of ID3 and  $k$ -NN-boosting

<i>Database</i>	<i>ID3 error</i>	<i>k-NN-boosting</i> <i>(best)</i>	<i>K value</i>	<i>k-NN-boosting</i> <i>(worst)</i>	<i>K value</i>	<i>Average</i> <i>(over all K)</i>
Diabetes	29.43 ± 0.40	<b>29.04</b> ± <b>1.78</b>	5	32.68 ±32.68	10	31.26 ± 1.37
Australian	18.26 ± 1.31	<b>17.97</b> ± <b>0.78</b>	6	19.42 ± 1.26	1	18.55 ± 0.32
Heart	27.78 ± 0.77	<b>21.85</b> ± <b>0.66</b>	1	27.78 ± 3.10	6	<b>25.48</b> ± <b>3.29</b>
Monk2	53.95 ±5.58	<b>43.74</b> ± <b>5.30</b>	4	<b>46.75</b> ± <b>0.73</b>	5	<b>45.09</b> ± <b>1.03</b>
Wine	7.29 ±0.53	<b>5.03</b> ± <b>1.69</b>	2	<b>5.59</b> ± <b>1.87</b>	1	<b>5.04</b> ± <b>0.06</b>
Zoo	3.91 ±1.36	<b>2.91</b> ± <b>1.03</b>	4	3.91 ±1.36	1	<b>3.41</b> ± <b>0.25</b>
Waveform-21	24.84 ±0.25	<b>23.02</b> ± <b>0.27</b>	5	25.26 ± 0.38	8	<b>24.22</b> ± <b>0.45</b>
Nettalk	25.96 ± 0.27	<b>25.81</b> ± <b>0.50</b>	7	26.09 ± 0.44	10	<b>25.95</b> ± <b>0.01</b>
Letter	11.66 ± 0.20	<b>11.47</b> ± <b>0.25</b>	2	11.86 ± 0.21	9	11.66 ± 0.02
Shuttle	0.02 ±0.11	0.02 ±0.11	any	0.02 ± 0.11	any	0.02 ±0.00

It can be seen that in three out of ten databases (Heart, Monk2 and Waveform-21) there are significance improvements under a confidence level of 95%, while no significantly worse performance is found in any database for any K value.

Let us observe that in several cases where no significant difference can be found, the mean value obtained by the new proposed approach outperforms ID3, as explained above.

In order to give an idea about the increment in the number of instances that this approach implies, in Table 4 the size of the augmented databases is drawn. The values appearing in the column labeled  $K = n$  corresponds to the size of the database generated from the entire original database when applying the first step of  $k$ -NN-boosting. As it can be seen, the size increase is not very high, and so it does not really affect to the computation load of the classification tree model induction performed by the ID3 algorithm.

$K$ -NN-boosting is a model induction algorithm belonging to the classification tree family, in which the  $k$ -NN paradigm is just used to modify the database the tree structure is learned from. Due to this characteristic of the algorithm, the performance comparison is done between the ID3 paradigm and our proposed one, as they work in a similar manner.

**Table 3.**  $K$ -NN-boosting vs. ID3 for every  $K$ . A  $\uparrow$  sign means that  $k$ -NN-boosting outperforms ID3 with a significance level of 95% (Wilcoxon test)

Database	$K=1$	$K=2$	$K=3$	$K=4$	$K=5$	$K=6$	$K=7$	$K=8$	$K=9$	$K=10$
Diabetes	=	=	=	=	=	=	=	=	=	=
Australian	=	=	=	=	=	=	=	=	=	=
Heart	$\uparrow$	=	=	=	=	=	=	=	=	=
Monk2	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	=	=	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$
Wine	=	=	=	=	=	=	=	=	=	=
Zoo	=	=	=	=	=	=	=	=	=	=
Waveform-21	=	=	=	=	$\uparrow$	=	=	=	$\uparrow$	=
Nettalk	=	=	=	=	=	=	=	=	=	=
Letter	=	=	=	=	=	=	=	=	=	=
Shuttle	=	=	=	=	=	=	=	=	=	=

**Table 4.** Sizes of the augmented databases

Database	Original size	$K=1$	$K=2$	$K=3$	$K=4$	$K=5$	$K=6$	$K=7$	$K=8$	$K=9$	$K=10$
Diabetes	768	1014	990	1003	987	987	976	977	973	972	969
Australian	690	928	916	916	909	905	895	893	894	897	890
Heart	270	385	375	365	360	360	364	359	360	363	366
Monk2	432	552	580	580	588	604	590	575	565	564	565
Wine	178	219	236	227	238	232	234	238	236	229	237
Zoo	101	103	123	108	106	109	111	113	117	120	122
Wavef.-21	5000	6098	6129	5930	5964	5907	5891	5851	5848	5824	5824
Nettalk	14471	15318	15059	15103	15065	15085	15069	15077	15056	15059	15061
Letter	20000	20746	20993	20799	20889	20828	20857	20862	20920	20922	20991
Shuttle	58000	58098	58111	58096	58108	58111	58112	58111	58120	58129	58133

## 6 Conclusions and Further Work

In this paper a new hybrid classifier that combines Classification Trees (ID3) with distance-based algorithms is presented. The main idea is to augment the training test duplicating the badly classified cases according to  $k$ -NN algorithm. The underlying idea is to test if one algorithm ( $k$ -NN) could be used to boost a different one (ID3).

The experimental results support the idea that such boosting is possible and deserve further research. A more complete experimental work on more databases as well as another weight changing schemas (let us remember that our approach is equivalent to double the weight of misclassified instances) could be subject of exhaustive research.

Further work could focus on other classification trees construction methods, as C4.5 [24] or Oc1 [21].

An extension of the presented approach is to select among the feature subset that better performance presents by the classification point of view. A Feature Subset Selection [11] [12] [26] technique can be applied in order to select which of the predictor variables should be used. This could take advantage in the hybrid classifier construction, as well as in the accuracy.

## 7 Acknowledgments

This work has been supported by the University of the Basque Country under grant 1/UPV00140.226-E-15412/2003 and by the Gipuzkoako Foru Aldundia OF-761/2003.

## References

1. Aha, D., Kibler, D., and Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning*, 6:37–66.
2. Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). *Classification and Regression Trees*. Monterey, CA: Wadsworth.
3. Cover, T. M. and Hart, P. E. (1967). Nearest neighbor pattern classification. *IEEE Trans. IT-13*, 1:21–27.
4. Cowell, R. G., Dawid, A. P., Lauritzen, S. L., and Spiegelhalter, D. J. (1999). *Probabilistic Networks and Expert Systems*. Springer.
5. Dasarathy, B. V. (1991). Nearest neighbor (nn) norms: Nn pattern recognition classification techniques. *IEEE Computer Society Press*.
6. Dietterich, T. G. (1997). Machine learning research: four current directions. *AI Magazine*, 18(4):97–136.
7. Freund, Y. and Schapire, R. E. (1999). A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5):771–780.
8. Gama, J. (2000). *Combining Classification Algorithms*. Phd Thesis. University of Porto.
9. Gunes, V., Ménard, M., and Loonis, P. (2003). Combination, cooperation and selection of classifiers: A state of the art. *International Journal of Pattern Recognition*, 17:1303–1324.
10. Ho, T. K. and Srihati, S. N. (1994). Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:66–75.
11. Inza, I., Larrañaga, P., Etxeberria, R., and Sierra, B. (2000). Feature subset selection by bayesian networks based optimization. *Artificial Intelligence*, 123(1-2):157–184.
12. Inza, I., Larrañaga, P., and Sierra, B. (2001). Feature subset selection by bayesian networks: a comparison with genetic and sequential algorithms. *International Journal of Approximate Reasoning*, 27(2):143–164.
13. Kohavi, R. (1996). Scaling up the accuracy of naive-bayes classifiers: a decision-tree hybrid. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*.
14. Lu, Y. (1996). Knowledge integration in a multiple classifier system. *Applied Intelligence*, 6:75–86.

15. Martin, J. K. (1997). An exact probability metric for decision tree splitting and stopping. *Machine Learning*, 28.
16. Martínez-Otzeta, J. M. and Sierra, B. (2004). Analysis of the iterated probabilistic weighted k-nearest neighbor method, a new distance-based algorithm. In *6th International Conference on Enterprise Information Systems (ICEIS)*, volume 2, pages 233–240.
17. Michie, D., Spiegelhalter, D. J., and Taylor, C. C. e. (1995). Machine learning, neural and statistical classification.
18. Mingers, J. (1988). A comparison of methods of pruning induced rule trees. *Technical Report. Coventry, England: University of Warwick, School of Industrial and Business Studies*, 1.
19. Mitchell, T. (1997). *Machine Learning*. McGraw-Hill.
20. Murphy, P. M. and Aha, D. W. (1994). Uci repository of machine learning databases.
21. Murthy, S. K., Kasif, S., and Salzberg, S. (1994). A system for the induction of oblique decision trees. *Journal of Artificial Intelligence Research*, 2:1–33.
22. Pearl, J. (1987). Evidential reasoning using stochastic simulation of causal models. *Artificial Intelligence*, 32(2):245–257.
23. Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1:81–106.
24. Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, Los Altos, California.
25. Sierra, B. and Lazkano, E. (2002). Probabilistic-weighted k nearest neighbor algorithm: a new approach for gene expression based classification. In *KES02 proceedings*, pages 932–939. IOS press.
26. Sierra, B., Lazkano, E., Inza, I., Merino, M., Larrañaga, P., and Quiroga, J. (2001a). Prototype selection and feature subset selection by estimation of distribution algorithms. a case study in the survival of cirrhotic patients treated with tips. *Artificial Intelligence in Medicine*, pages 20–29.
27. Sierra, B., Serrano, N., Larrañaga, P., Plasencia, E. J., Inza, I., Jiménez, J. J., Revuelta, P., and Mora, M. L. (2001b). Using bayesian networks in the construction of a bi-level multi-classifier. *Artificial Intelligence in Medicine*, 22:233–248.
28. Sierra, B., Serrano, N., Larrañaga, P., Plasencia, E. J., Inza, I., Jiménez, J. J., Revuelta, P., and Mora, M. L. (1999). Machine learning inspired approaches to combine standard medical measures at an intensive care unit. *Lecture Notes in Artificial Intelligence*, 1620:366–371.
29. Stone, M. (1974). Cross-validation choice and assessment of statistical procedures. *Journal Royal of Statistical Society*, 36:111–147.
30. Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics*, 1:80–83.
31. Wolpert, D. (1992). Stacked generalization. *Neural Networks*, 5:241–259.
32. Xu, L., Kryzak, A., and Suen, C. Y. (1992). Methods for combining multiple classifiers and their applications to handwriting recognition. *IEEE Transactions on SMC*, 22:418–435.

# The Scamseek Project – Text mining for Financial Scams on the Internet

Jon Patrick

Sydney Language Technology Research Group  
School of Information Technologies  
University of Sydney  
and  
Capital Markets Co-operative Research Centre  
[jonpat@it.usyd.edu](mailto:jonpat@it.usyd.edu)

## Abstract

The Scamseek project, as commissioned by ASIC has the principal objective of building an industrially viable system that retrieves potential scam candidate documents from the Internet and classifies them as to their potential risk of containing an illegal investment proposal or advice. The project produced multiple classifiers for different types of data, and achieved higher than expected performance statistics on classifications. The development of the system required the solution of two major problems in document classification, namely accurate identification of classes with very small footprints, <.1%, and classification using meaning intention rather than word strings. The approach taken used Systemic Functional Grammar to model the semantics of the scam classes and used unigrams with significant language pre-processing to assist in separating irrelevant documents. Litigations have been initiated by ASIC from classifications made by the system<sup>1</sup>. ASIC operates the system on a 24/7 basis. The estimate of savings in human effort in its monitoring role is the order of 100-fold. The estimate in savings to the community cannot be estimated readily but is likely to be of the order of tens of millions of dollars.

## 1 Introduction

Text Classification has a tradition of treating documents to be processed as a “bag-of-words” or n-grams, that is, the words or word groups within a text are treated as independent and uncorrelated with each other. Such a model of language is exceedingly simple but has been proven to satisfy many researchers.

The Scamseek project has sought to separate itself from the bag-of-words tradition of text classification. In particular the model of language used in the project was Systemic Functional Grammar (SFG) [1]. This model takes the position that language usage is a matter of choice set in a configuration of hierarchically layered strata of graphetics, graphology, lexicogrammar, semantics and context. Systemic grammar is a network of “systems” that interact with each other rather than a set of rules as with generative grammar.

In computing classifications, texts of a given class, apart from a small number of very common topic words, are more closely related by the minute intricacies of a weak network or chains of correlations that persist at low levels across small sub-sets of the class and the persistent meaning they represent, rather than by large persistent clusters of resoundingly dominant word sets that trumpet the presence of their class. In this case, the use of SFG states that the social context of the text’s composition dictates choices of meaning intentions which in turn influences the form of the text. The linguist’s task is to make sense of the decision making process and render it in a manner that might be suitable for computation. The computational linguist then has to convert the linguist’s model into a computable representation in the context of his target analytical methods which in this case is the procedures of machine learning.

---

<sup>1</sup> See ASIC Media Release 04-178: Grammax Investment Club operating unlicensed investment club is believed to have moved over \$10M overseas in the prior year.

## **2 Scamseek Project Specifications**

The Scamseek project was devised in two stages. The first stage had the aim of producing a production system for retrieving and classifying web pages. The client provided a manually classified corpus of about 8000 documents. The delivery time was 6 months from project commencement. The project team consisted of 1 linguist, 1 computational linguist and 3 software engineers.

The second phase ran for 9 months to 30 June 2004 and had the objectives of improving the accuracy of the web page classifier and the development of new classifiers for a number of other Internet data types. In Phase 2 the contract had more data sources to be scrutinized, entity recognition and performance requirements, plus in each case retrieval mechanisms had to be developed and for one source the corpus had to be compiled. The team was expanded with another linguist and computational linguist. Other part-time staff and consultants also made contributions.

## **3 Project Operations**

The Scamseek team was set up with a clear operational model that was effective throughout the life of the project, but adapted as work patterns developed to maturity. The operational model represented the task as consisting of 4 groups with different job functions; the client, linguists, computational linguists, and software engineers. The client was in contact with the linguists to deal with the classification of data. The linguists had the task of preparing the linguistic models of the data and passing that to the computational linguists who in turn had to prototype computational methods to compute the language models and devise machine learning experiments to optimize the classifiers. The computational linguists would pass their prototype code to the software engineers for efficient industrial quality implementation. This configuration operated effectively throughout the project development phases.

## **4 Computational Linguistic Research Topics**

### **4.1 Linguistic vs. administrative classes**

One of the early problems to emerge with the project requirements was the difference between the classification scheme of the client designed to conform to an administrative perception, that is, there are three types of scam under the law (unlicensed advisors, unregistered fundraising, and share ramping), and the linguistic manifestation of those three types. After a significant amount of linguistic analysis a set of registers (scam document sub-types based on their linguistic characteristics) were created representing subdivisions of the 3 scam types. This configuration was changed a number of times and expanded in phase 2 when the client opted to create different subdivisions in the data. The 3 scam types were treated as 1 document class with sub-classes or registers and the remaining part of the corpus was classified by the client into three more classes, Other-Agency-Scams, Scam-like and Irrelevant. These classes were also divided into registers to capture the linguistic variation within the classes. In all, over 50 registers were created with more than 20 in the scam class.

### **4.2 Linguists' compilation procedures**

The linguists conducted their work by a two part strategy. Firstly they read the documents and collated them into registers and at the same time created register descriptions. In the latter stages of the work the linguists were able to scrutinize documents that were incorrectly classified and attempt to adjust their ontologies for both the register of the misclassified document and the register it was computed to belong to.

### **4.3 Specification of linguistic model**

From the outset a decision was made to use a strong linguistic model to govern the direction of the work. This position was taken because the problem of identifying specialist content very thinly distributed and written in a particular manner was not believed accessible automatically by any other strategy.

The development of the linguistic model of the registers went hand in hand with the creation of the registers. The linguists read the documents and developed small scale characterizations of them. As the

work developed documents of similar ilk were paired together until all scams were assigned to a register and described for their features of differentiation and “scaminess”.

It was decided to represent register descriptions in an ontology rendered by XML. The upper part of these ontologies conformed to the SFG grammar as generally published, and the lower part is an ever increasing delicate rendition of the detail of the relevant content in the documents of the register. An objective of the work that was never achieved was the capacity to view a document and render it with an overlay of a register ontology and allow the linguists to do their extraction directly from the document image on the screen rather than their laborious hand collation.

The register descriptions and allocations resulted in a final list of more than 20 scam registers and 40 other registers spread across the 4 classes. At the same time the linguists with increasing understanding of the nature of the corpus advocated that greater amounts of the most structural components of the SFG model needed to be introduced into the assessment. Hence, the SFG networks for specific grammatical concepts were introduced as separate ontologies.

#### **4.4 Small footprints of target classes**

The scam class as a whole represented less than 2% of the corpus in phase 1, however with the development of the register model of the data there became registers with sizes <.1%. This represented significant problems with underrepresented classes and led to an experimental program to alleviate its effects. In phase 2 the client changes doubled the size of the scam class, however it also triggered a need to redevelop the whole set of scam registers to disperse a heterogeneous register into a homogeneous set. This also caused more small registers to be created and thereby not particularly improve the overall problem of the small footprints of registers.

Ultimately the small footprint problem was resolved by the development of the SFG ontologies for each register. The amount of effort spent on each individual register was related to some degree to the difficulty of separating it from other registers and therefore de facto addressed this problem.

#### **4.5 Hybrid Language Model**

The linguistic model can be considered to be designed in two parts. The first part was the register descriptions of the most important subdivisions of the corpus either on client needs basis, the scams, or for processing efficiency, that is, the largest groups of non-scam documents. The second part was the collection of all the non-scam classes and the completely irrelevant material which was the largest class (about 60%). These parts were in turn grouped into the four classes of the client. The task required was to develop classifiers for the major classes as well as the scam registers. The solution chosen was to develop ontologies for separating registers and use an n-grams approach to support the separation of the larger classes. This led to multiple lines of experimentation, namely, developing language processing functions for the ontologies, exploring the optimum feature selection for the classes independently of the registers, and, finally bringing the two solutions together to construct a combined classifier.

The SFG ontologies consisted of words and phrases from the texts organised in an SFG hierarchy, the upper parts reflect the theory of SFG and the lower parts represent the greater delicacy of the documents under analysis. The leaves of the ontologies were initially strings chosen from the texts classified in the given register. Over time the ontologies were developed and they became rich representations of the total document collection in the respective registers.

#### **4.6 Machine Learning – Classifier Development Programme**

The program for optimising the classifiers in the first phase concentrated on the problem of developing a single optimal classifier for web pages. In phase 2 separate classifiers were required for each data source and so experiments followed multiple strategies for all sources. SVMs were quickly identified as the best classifier for the data set.

Investigations were made of the selection of features from the collection of registers vis-à-vis the set of classes. While there was a significant overlap in the features chosen by an Information Gain metric there

was still an appreciable improvement by using the feature set chosen from the registers in the small classes and between the classes in the large classes thus giving a blend of feature selection methods.

Selection of features from register ontologies required an extensive series of experiments. The register ontologies performed well independently of other feature sets once they were developed to a very mature stage. Later the four grammatical ontologies were added which made various levels of contributions in intriguing ways. For example the Modality grammatical ontology performed particularly poorly by itself on some occasions classifying no documents correctly, yet when it was added to other models it consistently improved their scores. This result indicated clearly that there is an interaction effect within the grammatical ontologies that exploits a weak correlation not recognizable within the individual systems themselves. It is their union with other systems that created their strength. This result is entirely predictable with the SFG model of language and further justifies its use for this task.

#### **4.7 Mapping features to attributes.**

We use the terminology of *feature* for the linguistic phenomena that is the target of interest, and *attribute* for its numerical instantiation, and *mapping* for the computational transformation of the frequency count of the feature into its attribute representation. This distinction is unimportant for n-gram methods as the difference between features and attributes is inconsequential since the mapping transformation is trivial. This position cannot be taken in our work as the mapping transformation is different depending on the theoretical origin of the feature.

Feature representation for the ontologies was created by accumulating scores up the ontology tree. SFG in principle argues that the language is choice and therefore the important aspect of understanding the difference between two texts is the choice made by the authors. Hence by this principle the relative proportions of the choice to use one part of the tree over another should be the best differentiating feature. This is the case for the grammar ontologies but however does not apply to the register ontologies. The reason is that the register ontologies represent the most common semantic phenomena of a given register type, rather than choices between competing ways of expression. Hence, the attributes of domain register features are mappings to accumulative scores which are unnormalised, and grammar register features are mapped to proportional scores, whereas the n-gram word tokens are frequency counts normalized by document length.

## **5 Software Engineering Issues**

### **5.1 Regulating experimental practices.**

In the background, the engineers created an architecture that was intended to automate as much as possible the roll-out of the production system. As the production system required the use of the specific language processing methods and parameters of the very best machine learning experiment, the experimental programme had to be fully integrated into the engineers' software production process. Hence all computational linguists were coerced by the engineers into producing their code within the CVS system. This ensured that all the computational linguists' code was designed, at least architecturally, to fit the current production system.

### **5.2 Automatic roll-out of production classifiers.**

The integration of the computational linguists work into the CVS system ultimately enabled the complete automatic generation of the production system merely by supplying the number of the experiment which had produced the "best" classifier. With this number all the language models, all the language processing code and all the background system code (database schema, user interfaces, data retrieval, etc.) were automatically assembled into a single system for shipping to the client.

### **5.3 Use of Open Source Software**

The project used open source software for all aspects of its operations. The underlying operating system was Linux. Programming was in Python and interfaces were constructed using GTK with GLADE and CVS was used for code management and Bugzilla used for software revision requests. Postgres was used

for database management, and all machine learning experiments used the Weka suite. The only purchased software was XMLSpy to manage the descriptions of the SFG ontologies.

## 6 Results –Phase 1

The results of the first phase of the web page classifier for the scam classes as applied to an audit corpus have performance values of: Precision=.75, Recall=.41, and F-value=.53 and are to be contrasted with the laboratory results of the completed system on the training corpus using 10-fold cross validation of: Precision = .74, Recall = .35, F=.48. A baseline of 1000 single words has F=.21. ASIC was entirely satisfied with these results and made a commitment to a larger project in Phase 2.

This corpus was unseen by the development team and made available by ASIC at the time of delivery of the system. The processing was conducted by the ASIC staff and the project team was given one week in which to request revisions to ASIC's manual classifications. The Scamseek classifier in this instance identified 4 scams that had been manually misclassified by ASIC.

## 7 Results – Phase 2

The results of the second phase of the web page classifier for the scam classes applied to the corpus are presented in figure 1. ASIC was satisfied by the performance of the system in phase 1 not to require a second audit corpus assessment. Figure 1 provides results for 3 separate corpora, web pages as in the phase 1 experiments, and two other corpora developed for phase 2. The Web Pages result represents the system delivered to ASIC as of 30 June, 2004. The exact nature of the other corpora cannot be presented due to security obligations. The performance figures are determined by 10-fold cross-validation.

**Figure 1.** The performance results from the web pages classifier and 2 other classifiers for identifying scams on the Internet as delivered to ASIC.

	Web Pages	Corpus 2	Corpus 3
Precision	.744	.850	.852
Recall	.528	.834	.639
F-value	.618	.844	.730
Scam/non-scam texts	373/6391	686/1483	1395/13716

## 8 Conclusions

The Scamseek project is a success for ASIC in that it is operable 24 hours a day 7 days a week. In its first operational run it discovered an activity that has since been taken to the stage of litigation. The estimate of savings in human effort in its monitoring role is the order of 100-fold, as previously ASIC had to read 80 documents to find one of interest they now read 5 documents to find 4 of interest. The estimate in savings to the community by bringing speedier detection and intervention of scams cannot be estimated readily but is likely to be of the order of tens of millions of dollars. ASIC is not prepared to release all details about the technology but has released the following summary statement: "The Scamseek technology is deployed in

such a way that any scam proposal on any Internet channel that is generated in Australia or directed at Australians is highly likely to come under scrutiny”.

The research contribution has been significant in that it is the first project that has used Systemic Functional Grammar for automated text classification. Solutions to serious problems in practical text classification, namely unbalanced classes, and the integration of semantic and n-gram language models have also been developed.

The project has also made a significant contribution to the issues of software engineering in language technology in that it has shown that computational linguistics research can be performed in the context of reaching industrial objectives.

## **9 Acknowledgements**

The following people worked on the Scamseek project and made contributions to the final solutions, Michele Wong, Kathryn Tuckwell, Stephen Anthony, Tim Yeates, Dr. James Farrow, Neil Balgi, Jian Hu, Carlos Aya, Will Radford, Mathew Honnibal, David Smoker, Naomi Carter. The following doctoral students contributed to the work Maria Couchman, Casey Whitelaw, David Bell. The following people acted as advisors: Prof Christian Matthiessen, Prof Jim Martin, and Prof Vance Gledhill. Participating organizations were Australian Securities & Investment Commission (ASIC), Capital Markets Co-operative Research Centre (CMCRC), University of Sydney, Macquarie University, and the Australian Centre for Advanced Computing and Communications (AC3).

## **References**

1. Halliday, M. (1994). *Introduction to Functional Grammar*. 2<sup>nd</sup> Edition. London: Arnold.

# Fuzzy Document Filter for the Internet

Deepani B Guruge and Russel J Stonier

Faculty of Informatics and Communication, Central Queensland University,  
Rockhampton, QLD 4702, Australia  
{d.guruge,r.stonier}@cqu.edu.au

**Abstract.** Current major search engines on the web retrieve too many documents, of which only a small fraction are relevant to the user query. We propose a new fuzzy document- filtering algorithm to filter out documents irrelevant to the user query from the output of Internet search engines. This algorithm uses output of ‘Google’ search engine as the basic input and processes this input to filter documents most relevant to the query. The clustering algorithm used here is based on the fuzzy c-means with simple modifications to the membership function formulation and cluster prototype initialisation. It classifies input documents into 3 pre-defined clusters. Finally, clustered and context-based ranked URLs are presented to the user. The effectiveness of the algorithm has been tested using data provided by the eighth Text REtrieval Conference (TREC-8)[25] and also with on-line data. Experimental results were evaluated by using error matrix method, precision, recall and clustering validity measures.

## 1 Introduction

The amount of information on the Internet has exploded during the past decade but technologies that allow the full exploitation of the information on the Internet are still in their early stages. Several major search engines on the web retrieve both relevant and non-relevant material. Then the user has to search manually for relevant documents by traversing a topic hierarchy, into which a collection is categorised. As more information becomes available, it is a time consuming task to search for required relevant information [1]. Even now, users often find themselves having to wade through several hundred documents in response to their queries; this situation will only get worse in the future. To bridge this gap requires new data mining techniques and new processes that can be used for filtering information from the output produced by the search engines. An efficient and effective information filtering system can help users of the Internet to control inflow and satisfy their information needs [2].

In practice a user query may not be precisely defined. To deal with this ambiguity, it is helpful to introduce some ‘Fuzziness’ into the formulation of the problem [3]. Fuzzy logic application in information filtering has been used to obtain successful results for user queries [4],[5].

There has been much new research in the field of document retrieval over the last ten years. Sugimoto, Hori, and Ohsuga [6] introduced a document retrieval

system based on Automatic indexing techniques and statistical methods. Wang and Kitsuregawa [7], Thombros and Rijsbergen [8] used cosine similarity to calculate the similarity of a page with a cluster. In [9] the Jaccard coefficient is used to measure the similarity of two documents. The fuzzy k-nearest neighbour algorithm is implemented in [10] to classify documents into predefined clusters.

This research focuses on developing an effective document-filtering algorithm that classifies documents into 3 clusters, namely: ‘closely related’, ‘related’, and ‘not related’. It uses methodologies based on fuzzy clustering, automatic indexing and information retrieval techniques [11], [12]. Removal of stop words, stemming [13], and removal of HTML tags and comments are the initial steps of the process of assignment of index terms to the input documents. Then a fuzzy document clustering (FDC) algorithm which is based on the fuzzy c-means algorithm [3, 14] is used to classify input documents into appropriate clusters [15]. Finally Latent semantic Indexing (LSI) [17] is used to rank documents in the first two clusters based on their context.

We validate the effectiveness of the document filtering algorithm using data provided by the eighth Text REtrieval Conference (TREC-8)[25] and also with on-line data.

## 2 Web Document Filter

This section describes the modelling steps of the Fuzzy Document Filter (FDF). The system architecture given in Figure 1 describes the methodological design of the proposed filtering system. A FDC [15] which is a semi-supervised clustering algorithm is used to classify documents. Then classified documents will be presented to the next process (LSI) in which Context- based ranking is done.

### 2.1 Web Document Extractor (WDE)

The WDE (in Figure 1) uses Perl module WWW::Mechanize to search the Internet using the ‘Google’ search engine. After removing duplicate links (in Process 4 Figure 1) and filtering all the extracted document-links based on the meta-data that comes along with the links, the documents are downloaded into a temporary directory. Then all formatted documents (e.g. .pdf and .doc) are converted to text format. This is achieved by using the FileDetails.pm Perl Module. Files in this directory are automatically assigned name-tags and document numbers. These tags and document numbers are used to process these documents. Finally, details of the tags are mapped into the corresponding links. Off-line processing is used here to reduce document-accessing time.

### 2.2 Document Indexing

Each term in a document is then labelled with a document number ( $d_i$ ) and a weight ( $x_i$ ). A web document  $X^q$  with  $n$  key terms can therefore be represented as

$$X^q = ((d_i, t_1^q, x_1^q), (d_i, t_2^q, x_2^q), \dots, (d_i, t_n^q, x_n^q), ) \quad (1)$$

where

$d_i$  - is the document number given to each term in document  $X^q$   
 $t_i^q$  - is the  $i^{th}$  key term in document  $X^q$   
 $x_i^q$  - are the different weights assigned to terms in document  $X^q$   
 based on their frequencies.

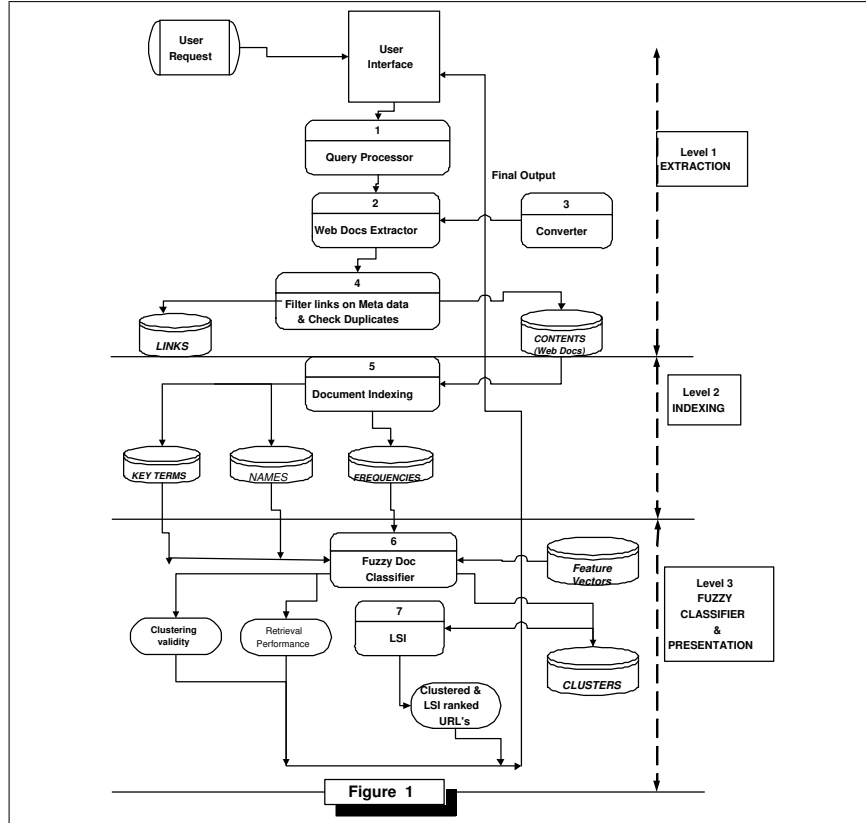


Fig. 1. Main System Architecture

**Selecting Key Terms.** Terms with low weights (low frequencies) are removed from the key term lists that are formed from the documents in the collection. This is achieved by defining a threshold which is dependent on the size of the document. The threshold is defined as shown below.

If (Doc-size <= 10 KB) then Threshold=1

```

else if (Doc-size >10 KB) and (Doc-size <40 KB) Threshold=2
else      Threshold=3

```

The system then provides two methods to enter another set of  $n_1$  key terms (including initial query or not) that can refine the initial output produced by the search engine. The system prepares a word list by selecting a few characteristic terms of high frequency from the top 20 documents in the search engine output. The user is able to use terms from this list and input  $n_1$  terms or the user can input their own key terms to the system.

### 3 Classifier Architecture

This section describes the FDC (Process 6) algorithm which is used to classify documents into predefined clusters, namely: ‘closely related’, ‘related’, and ‘not related’ which we shall refer to as cluster 0, 1 and 2 respectively. This FDC algorithm is based on fuzzy c-means (FCM) of Bezdek [3, 16] with modifications to the membership function formulation and cluster prototype initialisation [15]. To overcome problems encountered with FCM, a factor  $sw_{kq}$  was introduced to the membership function based on the number of user selected key terms appearing in the feature vector and cluster prototype (see Section 3.1). A further problem with FCM in application to document clustering [15] is that the resulting clusters are affected by the weights chosen to initialise cluster prototypes. We use evolutionary algorithms to validate an initialisation of weights or cluster prototype components, see Section 5.1.

#### 3.1 Modified Membership Function

The Factor  $sw_{kq} = \frac{\min(n_q, n_k)}{\max(n_q, n_k)}$  which lies in  $[0, 1]$ , measures the amount of overlap between the components of the feature vector and the cluster prototype, where  $n_q$  is the number of selected key terms appearing in document  $q$  and  $n_k$  is the number of components appearing in cluster  $k$ . When feature vector and cluster prototype both have the same number of features then the overlap  $sw_{kq}$  of 2 vectors is one. In order to make  $sw_{kq}$  more effective, initially we assign a lesser number of term-weights into the components of the second and third clusters as shown below. Note CN=Cluster Number and  $n_1$ =number of user selected terms. If  $CN = 0$ , initially  $n_k$  in Cluster 0 =  $n_1$ . If  $CN = 1$ , initially  $n_k$  in Cluster 1 =  $\text{int}(n_1/2) + 1$ . If  $CN = 2$ , initially  $n_k$  in Cluster 2 =  $\text{int}(n_1/2)$ . For example, if a particular document contains all most all the user-selected key terms, then  $n_q$  and  $n_k$  are high in Cluster 0 and the documents will be assigned a higher  $sw_{kq}$  in Cluster 0 than in Cluster 1 and 2. By multiplying  $\mu_{kq}^{FCM}$  with higher  $sw_{kq}$  we can assign a higher weight to documents with all the user-selected terms. This gives higher membership value for Cluster 0, rather than Cluster 1 and 2. The new membership function can be defined as:

$$\mu_{kq}^{new} = \frac{\mu_{kq}^{FCM} \times sw_{kq}}{\sum_{i=0}^{k_1-1} \mu_{iq}^{FCM} \times sw_{iq}} \quad (2)$$

In classical c-means algorithm membership function is defined as [7]

$$\mu_{kq}^{FCM} = \frac{\left(d_{kq}^{-2}(X^q, Z^k)\right)^{1/(p-1)}}{\sum_{k=0}^{k_1-1} \left(d_{kq}^{-2}(X^q, Z^k)\right)^{1/(p-1)}} \quad (3)$$

Where  $X^q = (x_1, \dots, x_{n_1})$  is a feature vector and  $Z^k = (z_1, \dots, z_{n_1})$  is a prototype vector for cluster  $k$ , both have dimension  $n_1$ .  $d_{kq}^2(X^q, Z^k)$  represents the Euclidean distance between the feature vector  $X^q$  and the cluster prototype  $Z^k$ .  $\mu_{kq}$  represents the degree of membership of feature vector  $X^q$  in the cluster  $Z^k$ ,  $p$  is any real number greater than 1 [3] [16]. By substituting  $\mu_{kq}^{FCM}$  in Equation (2) we have

$$\mu_{kq}^{new} = \frac{\left( \frac{\left(d_{kq}^{-2}(X^q, Z^k)\right)^{1/(p-1)}}{\sum_{k=0}^{k_1-1} \left(d_{kq}^{-2}(X^q, Z^k)\right)^{1/(p-1)}} \right) \times sw_{kq}}{\sum_{i=0}^{k_1-1} \left[ \left( \frac{\left(d_{iq}^{-2}(X^q, Z^k)\right)^{1/(p-1)}}{\sum_{k=0}^{k_1-1} \left(d_{iq}^{-2}(X^q, Z^k)\right)^{1/(p-1)}} \right) \times sw_{iq} \right]}. \quad (4)$$

### 3.2 New FDC Algorithm

1. Initialise all membership values ( $\mu_{kj}$ ) by using Equation (3). Concept prototypes are initialised before starting the algorithm, see Section 5.
2. Compute the new concept prototypes by using

$$Z^k = \sum_{q=0}^{q_1-1} \mu_{qk} X^q \quad \text{where } k = 0, \dots, (k_1 - 1) \quad (5)$$

3. Update  $\mu_{kj}^\ell$  to  $\mu_{kj}^{\ell+1}$  using

$$\mu_{kj}^{\ell+1} = \frac{\mu_{kq}^{FCM(\ell+1)} \times sw_{kq}}{\sum_{i=0}^{k_1-1} \left( \mu_{iq}^{FCM(\ell+1)} \times sw_{iq} \right)} \quad (6)$$

4. Set  $\ell = \ell + 1$ , if  $|\mu_{kj}^{\ell+1} - \mu_{kj}^\ell| < \epsilon$  stop; else go to step 2.

Finally the FDC assigns feature vector (documents) to a cluster which has maximum membership value in the final FDC results.

## 4 Content-Based Ranking

After passing through the classifier, documents in clusters 0 and 1 are presented to the next process LSI. Latent semantic indexing (LSI) uses truncated singular value decomposition (SVD) [17-20] to estimate the structure in word usage across documents and place documents with similar word usage patterns near each other in the term-document space. LSI starts with a terms (m) by documents (n) matrix A [17, 19]. LSI uses documents classified into Cluster 0 as training data. In order to construct matrix A document frequencies (*df*) of terms in the training corpus are calculated and terms with *df* less than predetermined threshold are removed. These terms that describe cluster 0 are presented to the user in descending order of importance to reformulate the query. Terms will not be shown in the list if they appear in less than 20% of the documents in cluster 0.

One of the common and usually effective methods for improving retrieval performance in vector methods is to transform the raw frequency of occurrence of a term in a document by some function. Such transformations normally have two components. Each term is assigned a global weight ( $G(i)$ ), indicating its overall importance in the document collection as an indexing term and also transform the term's frequency in the document which is called a local weighting ( $L(i, j)$ ) [17, 18, 21]. We can write Global and Local weighting as,

$$a_{ij} = L(i, j) \times G(i) \quad (7)$$

Results in [18] indicate a log transformation of the local cell entries combined with a global entropy (1-entropy) weight for terms is the most effective term-weighting scheme. In Local weighting  $\log(\text{Term\_frequency} + 1)$  takes the log of the raw term frequency, thus dampening effects of large differences in frequencies. Entropy (Global weighting) is based on information theoretic ideas and is the most sophisticated weighting scheme. The average uncertainty or entropy of a term is given by

$$\sum_j \frac{p_{ij} \log(p_{ij})}{\log(ndocs)} \text{ where } p_{ij} = \frac{tf_{ij}}{gf_i} \quad (8)$$

$tf_{ij}$  is the frequency of term  $i$  in document  $j$ ,  $gf_i$  is the total number of times term occurs in the whole collection,  $ndocs$  is the number of documents in the document collection. Subtracting the quantity in (7) from a constant assigns minimum weight to terms, which are concentrated in a few documents. Entropy takes into account the distribution of terms over documents [18].

Next, the matrix  $A$  is decomposed by using SVD, into three other matrices of special form [17],  $A = U\Sigma V^T$ . This is a form of factor analysis where one component matrix ( $U$ ) describes the original row entities as vectors of derived orthogonal factor values, another describes the original column entities ( $V$ ) in the same way, and third is a diagonal matrix ( $\Sigma$ ) containing scaling values such that when the three components are matrix-multiplied, the original matrix is reconstructed [17, 21]. The diagonal matrix contains the monotonically decreasing

singular values of  $A$ . the first  $k$  columns of  $U$  and  $V$  matrices and first  $k$  (largest) singular values of  $A$  are used to construct a rank- $k$  approximation to  $A$ ,  $A_k$ .

The idea is that the  $A_k$  matrix, by containing only the first  $k$  independent linear components of  $A$ , captures the major associational structure in the matrix and throws out noise. In this reduced model, the closeness of objects is determined by the overall pattern of term usage, so objects can be near each other regardless of the precise words that are used to describe them, and their description depends on a kind of consensus of their term meanings, thus dampening the effects of polysemy [18].

In the LSI model, queries are formed into pseudo-documents that specify the location of the query ( $q$ ) in the reduced term-document space. The pseudo-document can be represented by  $\hat{q} = q^T U_k \Sigma_k^{-1}$ . Where  $q$  is simply the vector of words in the users query, multiplied by the appropriate term weights in Equation 7 [17]. Once the query is projected into the term-document space, one of several similarity measures can be applied to compare the position of the pseudo-document to the positions of the terms or documents in the reduced term- document space. The query vector is compared to all document vectors, and the documents are ranked by their similarity (nearness) to the query. Cosine similarity measure between the query vector and document vector is used to measure the similarity of documents to the user query [17]. According to the results of similarity measures, links in clusters 0 and 1 are ranked and returned to the user.

#### 4.1 Choosing the Number of Dimensions

Dimension reduction analysis removes much of the noise, but keeping too few dimensions would loose important information [17, 18]. The dimensionality of the feature set needs to be reduced while the maximum amount of information and pattern in the data set is preserved.

In principal component analysis, the two guidelines for data reduction that are commonly used in practice, are the Kraiser criterion and the Scree test, [23]. The Kraiser criterion retains only factors with eigenvalues greater than one and the Scree test use a graphical method to select the number of factors. In this graphical method plotting eigenvalues it is required to find the place where the smooth decrease of eigenvalues appears to level off to the right of the plot. We found Scree test was more effective for this system and wrote a small piece of code to implement this criteria within the working code of the algorithm, so that off line calculations were not required.

## 5 Experimental Setup

This section describes the experimental settings we used to test the clustering validity of the filtering system. Data provided by the eighth Text REtrieval Conference (TREC -8) [25] and on-line data were used. Specifically, we used TREC-8 queries with their corresponding collections and relevance judgements

supplied by NIST accessors[26]. Sixteen (16) topics from the TREC topics 401-450 were randomly selected. They were: 402, 403, 406, 407, 410, 412, 414, 415, 419, 420, 425, 427, 429, 430, 431, 436. These topics were then converted into queries and ran against the TREC-8 web track (small web) 2 gigabyte, 250,000 document collection. For each topic we used a few relevant documents as training data and the context of the query was built with the training documents. These words were used to form matrix A in the LSI process. We set the stopping condition ( $\epsilon$ ) for FDC algorithm as 0.001.

### 5.1 Solution for Initial Z-Values by Evolutionary Algorithm

As stated previously we found that the resulting clusters were greatly affected by the initialisation of prototype centers. It was found from experiments that if we assign higher values to the components of the first cluster prototype we obtain feature vectors with higher components in the first cluster. We justify this assumption by the use of evolutionary algorithms as shown below.

An evolutionary algorithm (EA) [30], is used to learn the initial values for cluster prototypes. Each individual string in the evolutionary population, is to uniquely represent the entire set of cluster prototypes (Z-values). This can be achieved as follows. Each Z-value is uniquely represented by a real number within the range  $[0, 1]$ . The complete set of N, Z-values for all  $k_1$  clusters (where  $k_1 = 3$  for this application) can therefore be represented as a linear individual string a row vector of  $N = k_1 \times n_1$  weights,

$$\begin{aligned} \tilde{Z} &= [\tilde{Z}^1 \tilde{Z}^2 \dots \tilde{Z}^{k_1}] \\ \tilde{Z}^k &= [z_1^k z_2^k \dots z_{n_1}^k] \end{aligned}$$

where  $z_n^k$  is a real number in the range  $[0,1]$  for  $n = 1, \dots, n_1$  and  $k = 1, \dots, k_1$ .  $n_1$  and  $k_1$  are number of components in a cluster prototype and number of clusters respectively.

The initial population  $P(0) = \{\tilde{Z}_j : j = 1, \dots, M\}$ , where  $M$  is the number of strings (the population size), was determined by choosing the  $z_n^k$  as a random real number in  $[0, 1]$ . In determining successive populations a full replacement policy was used, tournament selection with size  $n_T$  (typically 2 or 4) was used to select parents in the current generation to produce children for the next generation. An elitism policy was also used with typically two (2) copies of the best string from a current generation passed to the next generation.

Mutation (with probability  $p_{mutation}$ ), was defined as a modified version of the Michalewicz mutation [30], using pseudo code similar to that below :

```
mutate=flip(pmutation); /* flip the biased coin
if (mutate){
    nmutation= nmutation + 1;
    pow=((1.0-padd) * (1.0-padd));
    fact=1.0 * (1.0 - power(MyRandom(),pow));
    if (flip(0.5))
```

```

        perturbation = fact * (lower_bound - allelevel);
    else
        perturbation = fact * (upper_bound - allelevel);
    temp = allelevel + perturbation ;
}
else
    temp = allelevel;
return (temp);

```

where  $MyRandom()$  is a procedure to generate a random real number in the range  $[0, 1]$  and for an individual  $Z_j(i)$ , upper-bound is 1, lower-bound is 0 and allelevel is  $Z_j(i)$ . The value of parameter  $padd$  is determined as follows.

```

paddout=gen_no/max_gen;
if (flip(0.5))
    padd = paddout;
else
    padd = 0.995;

```

Classical arithmetic crossover was used to form two children from two parent strings to be then added to the population in the next generation.

The fitness (objective) function for each string was simply determined as the objective function of FCM algorithm, [3]. It is

$$J(U, Z) = \sum_{q=1}^{(q_1)} \sum_{k=1}^{(k_1)} (\mu_{kq})^p ||\tilde{X}^q - \tilde{Z}^k||^2, \quad (9)$$

where  $||\tilde{X}^q - \tilde{Z}^k||^2$  represents the Euclidean distance between a feature vector  $\tilde{X}^q$  and a prototype  $\tilde{Z}^k$ ,  $\mu_{kq}$  represents the degree of membership of feature vector  $\tilde{X}^q$  in the cluster  $\tilde{Z}^k$ ,  $p$  is any real number greater than 1 and  $U = [\mu_{kq}]$  is a  $k_1 \times q_1$  fuzzy c-partition matrix. Here  $\{\tilde{X}^q : q = 1, \dots, q_1\}$  is a set of  $q_1$  feature vectors that is to be partitioned into  $k_1$  clusters. Feature vectors  $\tilde{X}^q = (x_1 \dots x_{n_1})$  and cluster prototype  $\tilde{Z}^k = (z_1 \dots z_{n_1})$  have dimension  $n_1$ , [3]. The objective function (9) uses the sum over the quadratic distances of the data to the prototypes, weighted with their membership degrees and it is to be minimised by the evolutionary algorithm.

The initial population is randomly generated, while ensuring that individual elements in the chromosome are within the range  $[0, 1]$ . We set length of the chromosome (N) as 30 ( $3 \times 10$ ) allowing the first 10 elements in the chromosome for initialising components of the first cluster prototype, 11 to 20 elements in the chromosome for initialising components of the second cluster prototype and 21 to 30 elements for initialising components of the third cluster prototype. From these 10 elements allocated for each cluster prototype only  $3 \times n_1$  components are used to calculate the fitness of each individual string in the population.

The fitness of each individual was modified to  $f_j = \delta J(U, Z)$ , where  $\delta$  denotes a scaling factor to increase the small values of  $J$  typically  $10^{-3}$  to values lying in the range  $[0 \ 100]$ . Once the new population is generated, the fitness of the

new population is evaluated and the fittest individuals are allowed to propagate through subsequent generations. A cross-over rate of 0.6 and mutation-rate of 0.05 was set. The fittest individual, the one with minimum fitness was examined after a preset number of generations.

Using the TREC-8 data, as test data, it was found for random initialisation of the population, that the fittest individual after some a few thousand generations, yielded Z-values for cluster 0 and cluster 1 higher (in the range 0.5 – 0.8) than the Z-values for cluster 2 (in the range 0.1 – 0.3).

This analysis justifies our procedure for prototype initialisation, setting values in cluster 0 and 1, higher than the values in cluster 2. For this research we set the values for cluster 0 and 1 in the range 0.6 – 0.8 and for cluster 2, 0.1 – 0.2.

## 6 Performance Evaluation

Retrieval effectiveness of the FDF was tested on Google output and the results were evaluated using, the error matrix method [31] and the two standard measures precision and recall [11]. The Performance of the FDC algorithm was tested on TREC-8 data and results were evaluated using precision and recall within the TREC-8 evaluation, as reported by NIST [26].

### Clustering Validity Measures

The Xie-Beni (XB) clustering validity measure has been used here to evaluate the clustering results. This measures the overall average compactness and separation of a fuzzy c-partition. Compactness and separation validity function XB is defined as the ratio of compactness  $\pi$  to the separation  $s$ , [29].

$$XB = \pi/s = \frac{\sigma/N}{(d_{min})^2} = \frac{\sum_{i=1}^N \sum_{j=1}^N \mu_{ij}^2 \|z_i - x_j\|^2}{n \min_{i,j} \|z_i - z_j\|^2} \quad (10)$$

A smaller XB indicates a partition in which all the clusters are overall compact and separate to each other.

In the relevance judgement given in the TREC-data, documents were classified into two classes *relevant* and *non-relevant*, using only titles in the TREC queries. Precision and recall values obtained by running FDC algorithm on TREC-8 data are shown in the Table 1 and Table 2. Graphical representation of the data in two Tables, Table 1 and Table 2 are given in Figure 2. Table 1 shows performance of the FDC algorithm relative to the documents filtered into the cluster 0 (*closely related*) and the Table 2 displays the performance relative to the documents filtered into the clusters 0 and 1 (*related*). In both tables the median performance of all the TREC-8 systems (pre-results) for corresponding queries are given [27]. The average precision relative to cluster 0 documents was 39% and the average precision relative to cluster 0 and cluster 1 was slightly low (33%). This is because the system classifies documents which are ‘related’ but ‘not closely related’ to the the given topic into the cluster 1. We selected

only 40 % of the top ranked (in Process 7) documents in the cluster 1. Average precision obtained for different systems for small web track (TREC-8) is in the range (2.9%-38%),[28]. Compared to these results retrieval effectiveness of the FDC algorithm is satisfactory. The average XB value obtain for this test data is low (0.03). We can say that all clusters are compact and well separated to each other.

Title terms given in the TREC-data were used as  $n_1$  terms input into the classifier. We can improve the performance of the FDC algorithm by reorganising the query based on the context built by the process LSI. By analysing this context (of the query), the user can input  $n_1$  key terms to the system.

**Table 1.** Precision and Recall Using only Cluster 0 Documents

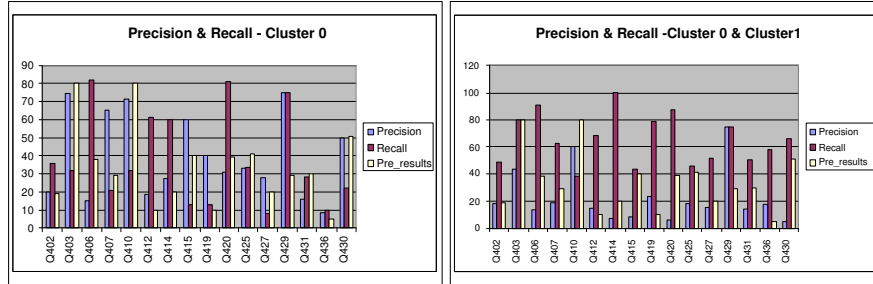
Query No	precision	Recall	TREC-8 Results Avg. Precision
Q402	20.0	35.7	25
Q403	74.4	31.9	53
Q406	15.3	81.8	30
Q407	65.2	20.5	28
Q410	71.4	31.9	80
Q412	18.3	61.4	10
Q414	27.3	60.0	20
Q415	60.0	13.0	40
Q419	40.0	13.0	10
Q420	31.0	81.3	29
Q425	32.7	33.3	40
Q427	27.8	8.1	20
Q429	75.0	75.0	30
Q431	16.0	28.0	30
Q436	8.3	9.7	5
Q430	50.0	22.2	50

The *Fuzzy Document Filter* (FDF) was applied to the output from the Google search engine for the initial query *Evolutionary Algorithms*. After removing duplicate links, non-accessible links and filtering all links based on meta data attached to each link, FDF downloaded 93 documents from 200 links in the Google output. (Only 200 links from the Google output were selected for testing.) These downloaded documents (93) were then filtered by using the secondary query *Evolutionary algorithms for optimization*. The filtered documents were classified into three classes: *closely related*, *related*, *not directly related* to the secondary query. The FDF classified 56 documents as *closely related*, 32 as *related*, 5 as *not directly related* to the secondary query. The value of XB was 0.034.

**Table 2.** Precision and Recall using Cluster 0 & Cluster 1 Documents

Query No	precision	Recall	TREC-8 Results Avg. Precision
Q402	18.1	48.9	25
Q403	43.2	80.2	53
Q406	13.5	90.9	30
Q407	18.6	63.0	28
Q410	60.0	38.3	80
Q412	15.0	68.6	10
Q414	7.1	100.0	20
Q415	8.9	43.5	40
Q419	23.0	79.0	10
Q420	6.3	87.5	29
Q425	18.3	45.8	40
Q427	15.4	51.6	20
Q429	75.0	75.0	30
Q431	14.0	50.0	30
Q436	18.0	58.0	5
Q430	5.0	66.7	50

For the purpose of evaluation five experts in the field Evolutionary Algorithms have been asked to evaluate the retrieved document set for the query *Evolutionary algorithms for optimization*. At the time of writing this paper only one evaluation form was received and the output of FDF was evaluated based on this report.



**Fig. 2.** Precision and Recall for TREC Data set

The error matrix method [31] was used to evaluate the clustering results. Overall Accuracy (OA) was computed by dividing the sum of the major diagonal elements by the total number of sample elements. OA is a measure of the total

match between reference and classification data. Accuracy of each individual category was measured by using producer's accuracy (PA) and user's accuracy (UA). PA is related to the error of omission which is calculated by dividing corresponding major diagonal elements by the total in reference data. UA is related to the errors of commission which is calculated by dividing corresponding major diagonal elements by the total in classification data. In the following table, Tables ,  $R^0$ ,  $R^1$ ,  $R^2$  are reference data and  $Z^0$ ,  $Z^1$ ,  $Z^2$  are classification data.

**Table 3.** Error Matrix for Downloaded Data Set

OA=0.70	$R^0$	$R^1$	$R^2$	PA	UA
$Z^0$	32	24	0	0.97	0.57
$Z^1$	1	31	0	0.55	0.97
$Z^2$	0	2	3	1	0.60

Examining the error matrix values, we can see that error matrix is diagonally dominant and the OA is high (0.7). Therefore we can say that, a high proportion of data items has been classified correctly. If we consider cluster 0 (closely related) values, condition of underestimation is minimum ( $PA^0 = 0.97$ ), but condition of overestimation is introduced in class 0 ( $UA^0 = 0.57$ ). Precision and Recall values obtained for on-line data are given in Table 4. The precision and recall values for all three clusters are greater than 50%. We can say that effectiveness of the FDF is satisfactory.

**Table 4.** Precision and Recall for Downloaded Data Set

Cluster No	precision	Recall
0	57	97
1	97	55
2	60	100

## 7 Conclusions

A fuzzy clustering algorithm for document filtering has been presented in this paper. In this algorithm a modified fuzzy c-means algorithm (FDC) is implemented to cluster documents into three predefined clusters namely: *Closely related*, *Related* and *Not related*.

To resolve a problem with initialisation of cluster prototypes for the fuzzy c-means algorithm, an evolutionary algorithm was used to gain an insight into

what values should be used to initialise the components of the cluster prototypes. It was found that the Z-values for cluster 0 and cluster 1 should be higher than the Z-values obtained for cluster 2. This validated what had been seen in simple experimentation.

Latent Semantic Indexing was implemented to rank documents in clusters 0 and 1 based on their context.

We evaluated the performance of the designed clustering system using TREC-8 data and also with output of Google search engine. It was shown that retrieval effectiveness of the designed system was satisfactory on the TREC-8 data set and also on the on-line data.

## Acknowledgements

Thanks to David Jones and the web development team of Faculty of Informatics and Communication, Central Queensland University for providing access to their Perl module (FileDetails.pm).

## References

1. S. Abuleil and M. Evens, "Building a machine -Learning system to categorize Arabic textJuly," presented at Eleventh international conference on intelligent systems:Emerging Technologies,, Boston, Massachusetts USA, 2002.
2. B. Sheth and P. Maes, "Information filtering using software agents," vol. 2002: MIT labs- software agents group, 1993-1994.
3. L. X. Wang, *A course in fuzzy systems and control*. U.S.A: Prentice Hall, 1997.
4. R. Chau and C. H. Yeh, "A fuzzy knowledge-based system for cross -lingual text retrieval," presented at Computational intelligence for modelling, control and automation, 1999.
5. C. H. Oh, K. Honda, and H. Ichihashi, "Fuzzy clustering categorical multivariate data," presented at Joint 9th IFSA word congress and 20th North American Fuzzy Information Processing Society (NAFIPS) International Conference, Vancouver, Canada, 2001.
6. M. Sugimoto, K. Hori, and S. Ohsuga, "A document Retrieval System for assisting creative research," presented at 3rd International Conference on Document Analysis & Recognition, Montreal, Canada, 1995
7. Y. Wang and M. Kitsuregawa, "Evaluating contents-link coupled web page clustering for web search Results," presented at Eleventh international conference on Information and knowledge management, Virginia, USA, 2002.
8. A. Thombras and C. J. Van Rijsbergen, "Query-sensitive similarity measures for the calculation of inter-document relationships," presented at Proceedings of the international conference on information and knowledge management, Georgia, USA, 2001.
9. T. H. Heveliwala, A. Gionis, D. Klein, and P. Indyk, "Evaluating strategies for similarity search on the web," presented at Eleventh international conference on world wide web, Honolulu, Hawaii, USA, 2002.

10. R. Chau and C. H. Yeh, "Building a Concept-based multilingual text retrieval system using fuzzy clustering and classification," presented at International Conference on Intelligent Agents, Web Technologies and Internet Commerce- IAWTIC, U.S.A, 2001.
11. G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. Singapore: McGraw-Hill International Book Company, 1984.
12. C. J. Van Rijsbergen, *Information Retrieval*. London: Butterworths, 1979.
13. M. F. Porter, "An Algorithm for Suffix Stripping," vol. 14, 1980, pp. 130- 137.
14. L. Zhang, "Comparison of Fuzzy c-means Algorithm and New Fuzzy Clustering and Fuzzy Merging Algorithm" University of Nevada, Reno, NV89557 2001.
15. R. J. Stonier and D. B. Guruge, "Building an Efficient Document Retrieval System Using Fuzzy Clustering," presented at First Indian International conference on Artificial Intelligent (IICAI), Hydreabad, India, 2003.
16. C. G. Looney, "Interactive Clustering and Merging with a new expected value," vol. 2002: University of Nevada, Pattern recognition Society, Reno, NV 89557, 2002.
17. M. W. Berry, S. T. Dumais, and G. W. O'Brien, "Using Linear Algebra for Intelligent Information Retrieval," SIAM Review, vol. 37, pp. 573-595, 1995.
18. S. T. Dumais, "Improving the retrieval of Information from external sources," *Behaviour Research Methods, Instruments and Computers*, vol. 23, pp. 229- 236, 1991.
19. S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by Latent Semantic Analysis," *Journal of the American Society for Information Science*, vol. 41, pp. 391-407, 1990.
20. T. K. Landauer and S. T. Dumais, "A Solution to Plato's Problem: The Latent Semantic Analysis Theory of the acquisition, induction, and representation of knowledge," *Psychological Review*, vol. 104, pp. 211-240, 1997.
21. T. K. Landauer, P. W. Foltz, and D. Laham, "An Introduction to Latent Semantic Analysis," *Discourse Processes*, vol. 25, pp. 259-284, 1998.
22. T. K. Landauer, D. Laham, and P. W. Foltz, "Learning Human-like Knowledge by Singular Value Decomposition: A progress Report," *Advances in Neural Information Processing Systems*, vol. 10, pp. 45-51, 1998.
23. STATISTICA, "Factor Analysis," vol. 2003: StatSoft, Inc, 1984-2004.
24. E. Binaghi, P. A. Brivio, P. Ghezzi, and A. Rampini, "A fuzzy set-based accuracy assessment of soft classification," *Pattern Recognition letters*, vol. 9, pp. 935-948, 1999.
25. E. M. Voorhees, D. Harman, "The Eighth Text REtrieval Conference (TREC-8)", Gaithersburg, Maryland, November 16-19,1999.
26. <http://trec.nist.gov/>
27. A. Berger, J. Lafferty, "The Weaver system for Document Retrieval", presented at The Eighth Text REtrieval Conference (TREC-8), Gaithersburg, Maryland, November 16-19,1999.
28. D. Hawking, E. M. Voorhees, N. craswell, P. Bailey, "Overview of the TREC-8 Web Track", presented at The Eighth Text REtrieval Conference (TREC-8), Gaithersburg, Maryland, November 16-19,1999.
29. X.L. Xie, G. Beni, "Validity Measures for Fuzzy Clustering", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 841-847, 1991.
30. Z. Michalewicz, "Genetic Algorithms Data Structures Evolution Programs" ,2nd Ed., Springer Verlag, (1994).
31. E. Binaghi, P. A. Brivio, P. Ghezzi, and A. Rampini, "A fuzzy set-based accuracy assessment of soft classification" *Pattern Recognition letters*, vol. 9, pp. 935-948, 1999.



# Informing the Curious Negotiator: Automatic News Extraction from the Internet

Debbie Zhang and Simeon J. Simoff

Faculty of Information Technology,  
University of Technology, Sydney  
Broadway PO Box 123, NSW 2007, Australia  
{debbiez, simeon}@it.uts.edu.au

**Abstract.** In negotiation, information acquisition and validation play an important role in the decision making process. In this paper we briefly present the framework of a smart data mining system for providing contextual information from the Internet to a negotiation agent. We then present one of its components in more details - an effective automated technique for extracting relevant articles from news web sites, so that they can be used further by the mining agents. Most current techniques experience difficulties to cope with changes in websites structure and formats. The proposed extracting process is completely automatic and independent of web site formats. The technique is based on identifying regularities in both format and content of the news web sites. The algorithms are applicable to both single- and multi-document web sites. Since invalid URLs can cause errors in data extraction, we also present a method for the negotiation agent to estimate the validity of the extracted data based on the frequency of the relevant words in the news title. This paper also presents a new procedure for constructing news data sets of given topics. The extracted news data set is further utilised by the parties involved in negotiation. The information retrieved from the data set can support both human and automated negotiators.

## 1 Introduction

The *curious negotiator* [1] is a multiagent system of competitive agents supporting multi-attribute negotiation where the set of issues is not fixed [2]. The overall goal of its design is to exploit the interplay between contextual information [3] and the development of offers in negotiation conducted in an electronic environment. Current design is illustrated in Figure 1. Negotiation agents apply the negotiation strategies in the negotiation process [4]. With respect to the curious negotiator the term ‘negotiation strategies’ includes strategies for developing the set of issues in an offer as well as *identifying, requesting and evaluating contextual information* including determining what information to table as the negotiation proceeds [5]. A negotiation strategy should generally rely on information drawn from the context of the negotiation. The significance of information to the negotiation process was analysed formally in the seminal paper by Milgrom and Weber [6] in which the Linkage Principle, relating the revelation of contextual information to the price that a purchaser is prepared to pay, was introduced. “Good negotiators, therefore, undertake integrated processes of knowledge acquisition that combine sources of knowledge obtained at and away from the negotiation table. “They learn in order to plan and plan in order to learn” [7]. The grand vision for curious negotiator encapsulates this observation. The mediation agents (labelled as ‘mediator’ in Figure 1) assist negotiation agents in the negotiation process. The role of observer agents (labelled as ‘observer’ in Figure 1) is to observe and analyse what is happening on the ‘negotiation table’ and to look for opportunities particularly from failed negotiations.

Successful negotiation relies on an understanding of how to ‘play’ the negotiation mechanism [5] and on contextual information. From a process management point of view, negotiation processes are interesting in that they are knowledge-driven emergent processes that can be fully managed provided that, first, full authority to negotiate is delegated to the agent and, second, sufficient contextual information can be derived from the market data, from the sources, available on the Internet (news feeds, company white papers, specialised articles, research papers) and other sources by the data mining bots. The dashed lines in Figure 1 contour two scenarios: “SA” – a semi-automated scenario in which the human agent receives and processes contextual information and affects the strategies of the negotiation agent, and “A” in which contextual information is distilled and passed to the negotiation agent in a form of parameters that are taken in consideration by the negotiation strategies. The curious negotiator is designed to incorporate data mining and information discovery methods [8] that operate under time constraints, including methods from the area of topic detection and event tracking research [9]. The idea is encapsulated in the “smart data miner” in Figure 1. The architecture of this specialised data mining system, which operates in tandem with the human or/and negotiation agent, is shown in Figure 2. Initially the information is extracted from various sources including on-line news media, virtual communities, company and government web sites. Extracted information is converted to a structured representation and then both representations are stored in the mining base. They are used for further analysis by different data mining algorithms, including different text and network mining agents. The ‘Source profile base’, includes a collection of time-stamped data about the behaviour of the approached sources like response time, the number of answered requests, dates when a new layout appears, redirections of requests, types of errors, subscription price, change in subscription price, change of the

level of service provided and other parameters. The ‘Source evaluator’ provides a number of estimates, e.g. ‘hold-up’, ‘reliability’, ‘cost’, ‘trust’ that evaluate the quality of the data sources from which the patterns have been extracted. These estimates are derived from the related data in the source profile base. This paper is limited to the techniques that cover the automatic extraction of relevant news articles. Regardless of whether we deal with scenario “SA” or “A”, there are a number of challenges in real world negotiations that the smart data mining system needs to address, including (i) critical pieces of information being held in different repositories; (ii) non-standard formats; (iii) changes in formats at the same repository; (iv) possible duplicative, inconsistent and erroneous data. This paper addresses the first three challenges in the context of providing news to the negotiation table. The scope of the paper covers the universal news bot shown within the dashed rectangle in Figure 2. The techniques considered in this paper are applicable for both scenarios, however, the details are beyond the scope of the paper.

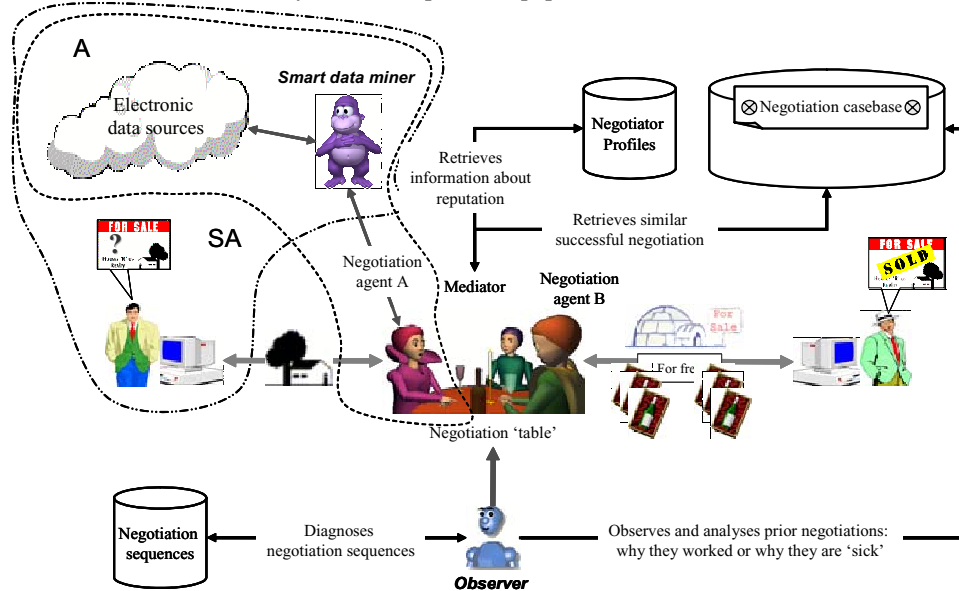


Figure 1. Current design of the curious negotiator (includes negotiation agent, mediator, observer and the smart data miner).

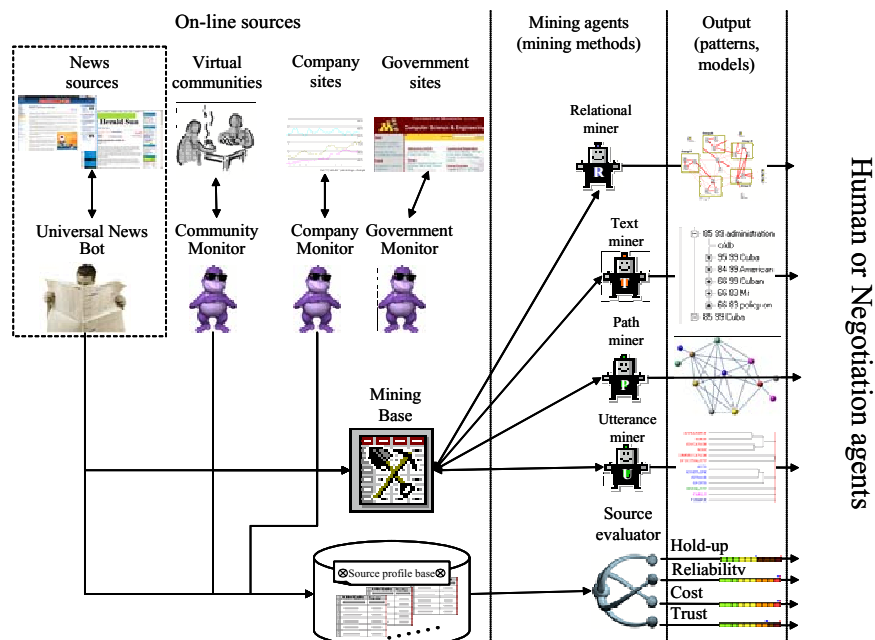


Figure 2. Smart data mining system for supporting negotiation with contextual information.

### 1.1 On-line News Media

Obtaining and verifying information from on-line sources takes time and resources. To reduce the impact of some delay factors on the net, the architecture of the data mining system in Figure 2 allows not only just-in-time operation, but also ‘pre-fetching’ some of the information that is expected to be necessary for a scheduled negotiation. In the context of news mining, the news bots fetch the news, which then are transformed into a structured form and both the structured and unstructured data are stored in the mining base (see Figure 3) for accessing by the mining agents (The fragment selected illustrated in Figure 3 shows only the text mining agent).

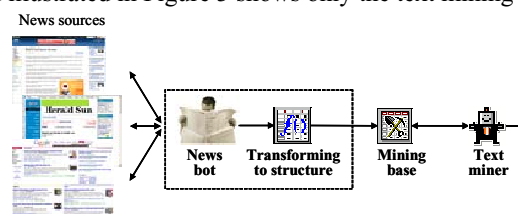


Figure 3. The news mining portion of the system

The focus of this paper is on the first phase – the automation of obtaining news from Internet sources. The news sources on the Internet include the websites of major news papers. The development of algorithms for finding the correct URLs that contain the requested news articles is within the scope of the intelligent crawler research. Major search engines, including Google (shown in Figure 4) and Yahoo recently provided a new functionality for news searching with user provided keywords. These news portals provide convenient interface for humans; answering queries in way similar to conventional search engine interface (see Figure 5).



Figure 4. News search engines web interfaces

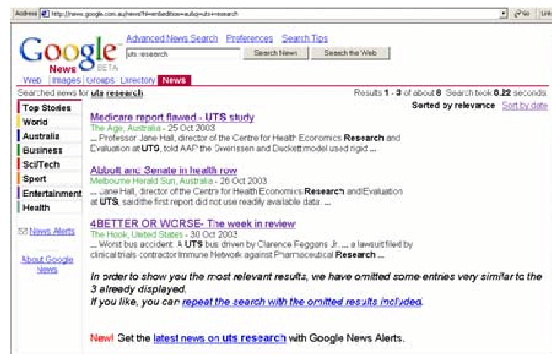


Figure 5. News search engines web interfaces – response to a query.

Within the framework of curious negotiator, a generic news bot should be able to retrieve automatically, classify and store the news article obtained by a search engine in an efficient way that can be further used by the other mining agents. The initiation of the process in just-in-time mode can be by the negotiation agent (scenario “A” in Figure 1) or the human player (scenario “SA” in Figure 1). In pre-fetch mode, a source monitoring agent is subscribed to the email digests that these sources distribute [10]. These sources include 2-3 sentences news abstract and the corresponding URLs for retrieving the full articles. The trigger for fetching an article can be a negotiation scheduler, using as initial information the topic, the list of items and the description of participants.

However, the automatic retrieval by a computer program of an individual news article from the URL that is obtained either from search results or from the pre-fetched list is a tedious job since the news content can come from different web sites. Different news sources have different layout and format as illustrated by the two examples of news websites in Figure 6 and Figure 7. The layout may vary from time to time even in the news coming from the same source. Hence when automating news retrieval, even for the same news site, it is impractical to develop a static template, as it will stop working when the layout is changed. It is even more impractical (if not impossible) to develop a predefined program (template) for each news web site in the whole Internet. In this paper we present a more generic approach to retrieve news articles regardless the web site format and bringing them to the smart data mining system of the curious negotiator.



Figure 6. A version of SMH news site format



Figure 7. A version of Herald Sun news site format

## 2 The Universal News Bot approach

Data extraction from Web documents is usually performed by software modules called wrappers. As explained in the previous section, hard-coded wrapper by using static template is tedious, error-prone and difficult to maintain. To overcome this difficulty, significant research has been done in the area of wrapper induction, which typically applies machine learning technology to generate wrappers automatically [11, 12]. WIEN is the first wrapper induction system that defined six wrapper classes (templates) to express the structures of web sites [13, 14]. STALKER - a wrapper, more efficient than WIEN [15], treats a web page as a tree-like structure and handles information extraction hierarchically. Gao and Sterling [16] have also done significant work on knowledge-based information extraction from the internet. However, most of the earlier wrapper techniques were tailored to particular types of documents

and none is specific for news content retrieval. The more recent techniques aim on data extraction from general semi-structured documents. The application of general content identification and retrieval methods to news data brings unnecessary overhead in processing. This paper proposes a technique that takes in account the characteristics of news web pages. Without loss of generality, the approach improves the processing efficiency and requires neither user specified examples or priori knowledge of the pages.

## 2.1 The data extraction method

The data extraction process is divided into three stages. The logical structure of the tagged (in our case, HTML) file is firstly identified and the text, which is most likely to be the news article, is extracted. During the second stage a filter is dynamically built and some extra text is filtered out if multiple documents from the same web site are available. During the third stage extracted data is validated by the developed keyword based validation method. The details are presented below.

### 2.1.1 Stage 1: Identifying the logical structure of the tagged file

News pages normally not only contain the news article, but more often, also related news headings, the news category, advertisements, and sometimes a search box. Although each web site may have a different format, web pages can always be broken down into content blocks. The layout in which these content blocks are arranged varies considerably across sites. The news article is expected to be the content block which is displayed on the “centre” of the page. Therefore, it is reasonable to assume that *the biggest block of text on the news web page is the news article*. Similar to McKeown et al.’s [17] approach, the biggest block of text is detected by counting the number of words in each block.

Most of web sites employ visible and invisible tables in conjunction with Cascading Style Sheets (CSS) to arrange their logical structures by using HTML table tags [18]. Table is designed to organize data into logical rows and columns. A table is enclosed within the `<table></table>` tag. Nested tables are normally used to form a complex layout structure. It is common for news web sites to display advertisements within news articles to attract reader’s attention. This is normally done by inserting nested tables that contain advertisements and other contents in the table that contains the news article. The pseudo code of the process is presented in Figure 8.

```

Input: HTML file
Output: The largest body of text contained in a table
Begin
1. Break down the HTML file into a one dimensional array, where
   each cell contains a line of text or an HTML tag
2. Remove the HTML tags except <table> and </table>
3. Set table_counter to 0
4. For each cell in the array:
   a. if <table> tag is encountered, increase table_counter by 1
   b. if </table> tag is encountered, decrease table_counter by 1
   c. if it is a text element, append it to the end of con-
       tainer[table_counter]
5. Return container[i] that contains the largest body of text by count-
   ing the number of words.
End

```

Figure 8. Pseudo code of the algorithm for identifying the largest text block.

### 2.1.2 Stage 2: Building internal filters dynamically

Although most of news web sites use tables for partitioning content blocks, there are some web sites that use other methods. Also, even for the web pages that use tables as the partition method, the table with the news article may contain a few extra lines of text at the beginning or the end of the article. Therefore, extraction accuracy can be improved by developing algorithms that do not rely on table tag information. Many web sites use templates to automatically generate pages and fill them with results of a database query, in particular, for news web sites. Hence, news under same category from same source is often with same format. When two or more web pages from same source become available, a filter can be constructed by comparing the extracted text from these pages. The filter contains the common header and tail of the text. The text is compared sentence by sentence from the beginning and the end between two files. Common sentences are regarded as part of web page template. Therefore, they should be removed from the file. The pseudo code of the process is shown in Figure 9.

Once the filter is generated, text is refined by removing the common header and tail text in the filter. Since the filter is dynamically generated, it is adjusted automatically when the web site format is changed.

**Input:** two text files from the same web site, each contains a news article

**Output:** a data structure contains:

String *URL*

String *Header*

String *Tail*

1. Remove all the html tags in the files.
  2. Break down the files into one dimensional arrays (a and b), each cell contains a line of text.
  3. For each cell of the array from beginning
    1. if  $a[i] == b[i]$ , append  $a[i]$  at the end of *Header* string
    2. if  $a[i] != b[i]$ , break;
  4. For each cell of the array from the end
    1. if  $a[i] == b[i]$ , insert  $a[i]$  at the beginning of *Tail* string
    2. if  $a[i] != b[i]$ , break
  5. Set the *URL* value to the common part of the URLs of two text file
- Return the data structure that contains *URL*, *Header* and *Tail*.

Figure 9. The pseudo code of dynamic filter generation.

### 2.1.3 Stage 3: Keyword based validation

Incorrect and out of date URLs can cause errors in the results of data extraction. Such errors can not be identified by the data extracting methods described in the previous sections. A simple validation method based on keyword frequency is developed to validate the data retrieved by the algorithms in Figure 8 and Figure 9.

The basic assumption is that a good news title should succinctly express the article's content. Therefore, the words contained in the news title are expected to be normally among the most frequent words appearing in a news article. Consequently, the words from the news title (except the stop words, which are filtered out) are considered as keywords. For situations when the news title is not available at the time of text extraction, the words in the first paragraph of the extracted data are considered as keywords, based on the assumption that title is always placed at the beginning of an article. The extracted text is regarded as the requested news article if it satisfies the following condition:

$$\min \left( w_1 \frac{l_t}{l_m}, w_2 \frac{n_k}{t_k}, w_3 k_f \right) > th \quad (1)$$

where:

$l_t$  total length

$l_m$  minimum length (predefined)

$n_k$  number of keyword that appears in the text at least once

$t_k$	total number of keywords
$k_f$	average keyword frequency
$w_1, w_2, w_3$	weighting values
$th$	threshold value (predefined)

The first term in equation 2.1 considers the total length of the extracted text. If the text length is unreasonably short, the text is unlikely to be a news article. The second term in the equation represents the percentage of the keywords that appeared in the text. The third term in the equation stands for the average frequency of the keywords that appeared in the text. The validation value takes the minimum value of these three and then compares with a predefined threshold to validate if the extracted text is the news article.

### 3 News Data Set Construction

The news data set for a given specific topic that will be used as the information sources for the negotiation table is dynamically constructed from on-line news articles. In stead of simply using keyword searching, the data mining agent constructs the news data set according to the concept related to the given keywords.

Similar to using searching engine, the negotiation agent provides a phrase or several keywords to the data mining agent to define the topic of the news it requests. The data mining agent submits the query to a news searching engine. In general, large amount of searching results are returned. The data mining agent only retrieve the most relevant data evaluated by the keyword frequency and their proximity position. Based on the assumption that a concept can be represented by a set of keywords, which occur frequently inside particular collection of documents, the most frequent keywords (terms) from the retrieved data set are extracted and considered to be related to the same concept. The extracted keywords are resubmitted to the search engine. The process of query submission, data retrieval and keyword extraction is repeated until the search results start to derail from the given topic. The news articles used in this section are extracted from HTML files by the algorithms described in section 2.

#### 3.1 Key Phrase Extraction

As it is introduced in the previous section, key phrase extraction plays an important role in the data set construction process in this project. Many studies have been conducted in the area of automatic keyword generation from text documents. Most of these methods are based on syntactic analysis using statistical co-occurrence of word types in text and vector space representation of the documents [19]. Hulth [20] suggested the quality of keywords that generated by frequency analysis was significantly improved when a domain specific thesaurus is used as a second knowledge source.

Therefore, a similar approach that employs a domain specific thesaurus in the key phrase extraction process is adopted in this project. Since the frequently used words represent the topic of a document in greater degree than less frequently used words, the frequency of the words and phrases predefined in the domain thesaurus that appear in the documents are calculated and the top ranked words or phrases are considered as the keywords.

There are many publications on automatic thesaurus construction. We applied a relative simple approach based on word frequency count. The news articles in many news web sites are organized into the categories of: World, National, Business, Science (Technology), Sport and Entertainment. To build the domain specific thesaurus, a large number of news articles under each category are collected, and each of such categories represents a domain. Figure 10 shows the steps of building the database of key phases for each category (the domain thesauruses). Word stemming problem was resolved by using a simple stemming algorithm that two words are considered to have the same stem if they have the same beginnings and their endings differ in one or two characters [21]). Stop words are not counted in each document.

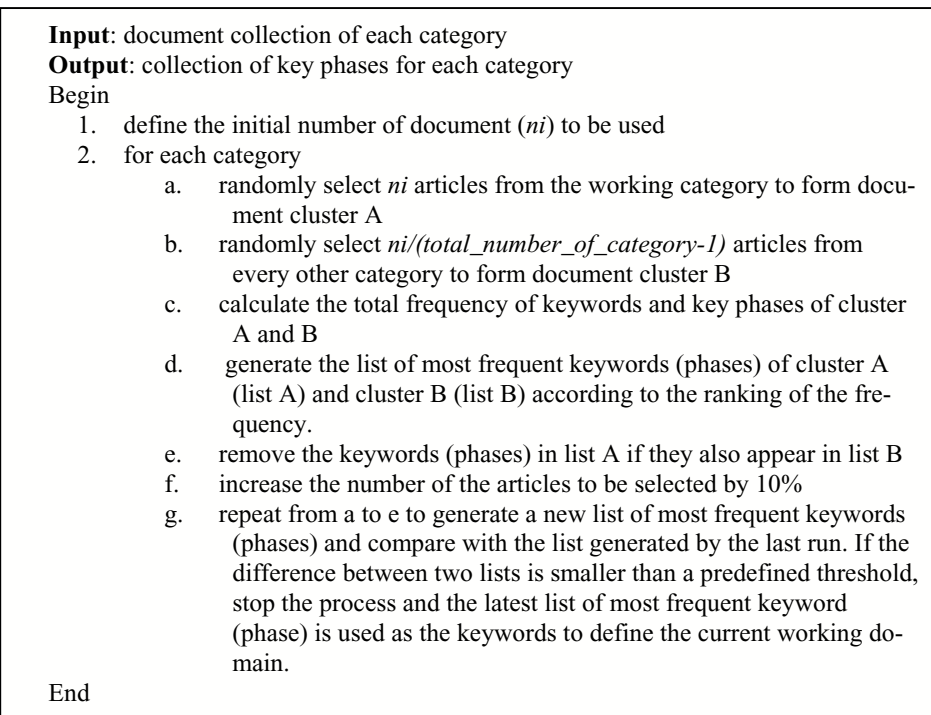


Figure 10. The pseudo code of domain thesaurus construction.

In 2.c of the above process, a sequence of words is defined as a phrase if it satisfies:

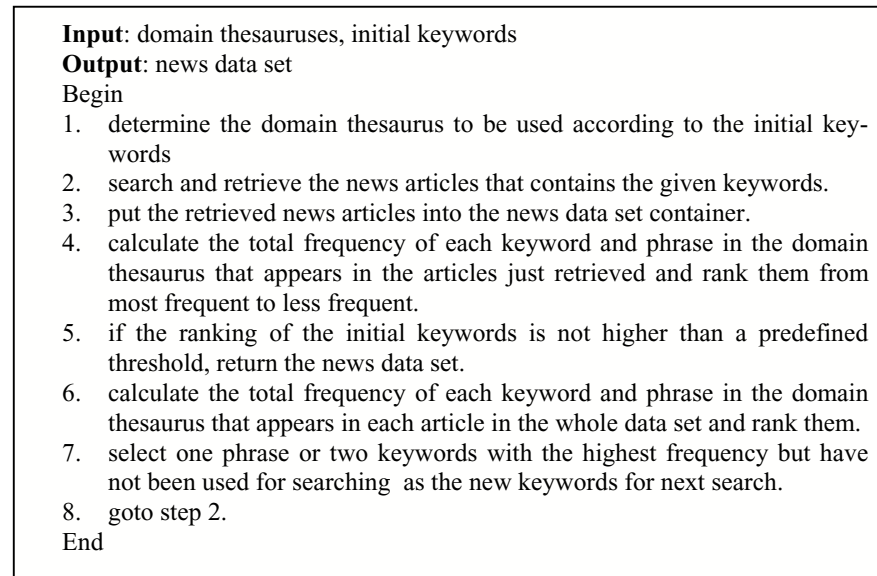
$$\frac{f_{seq}}{f_{average}} > th \quad (2)$$

where:

- $f_{seq}$  frequency of the sequence of words that appears in the same sentence in the whole cluster
- $f_{average}$  average frequency of each word in the sequence.
- $th$  threshold value (predefined)

### 3.2 News Data Set Construction process

The news data set is constructed by repeating the news retrieval and keyword extraction process. Figure 11 shows the detail procedure of the construction process. The news data domain is determined by searching the initial provided keywords (phase) in the domain thesauruses.



**Figure 11. The pseudo code of news data set construction for a given concept**

## 4 Experimental Results

Experiments have been conducted in two steps: first to evaluate the news extraction algorithm. Second, a news data set was constructed and manually examined.

### 4.1 News extraction

The proposed methods of extracting news articles were evaluated by the experiments using some of the most popular Australian and International news web sites, which are shown in Table 1. 200 pages from each URL location were tested. The average process time for each page was 436 milliseconds on a Pentium 4 1.60 GHz computer. The notions used in the table are explained below:

- *Correct* – on average 0% error rate in the extracted text of a single web page;
- *Minor Error* – on average less than 5% error rate in the extracted text of a single web page;
- *Major Error* – on average between 5% to 30% error rates in the extracted text of a single web page;
- *Error* – on average more than 30% error rate in the extracted text of a single web page

Table 1. News sites for testing the news article extraction algorithm and the results

URL Location	Accuracy [without Filter]	Accuracy [with Filter]
www.smh.com.au/national	Minor Error	Correct
www.smh.com/business	Minor Error	Correct
www.usatoday.com/news/world	Minor Error	Minor Error (Error Rate Reduced)
www.usatoday.com/news/nation	Minor Error	Minor Error (Error Rate Reduced)
http://abcnews.go.com/sections/us	Minor Error	Correct
http://abcnews.go.com/sections/world	Minor Error	Correct
http://money.cnn.com	Correct	Correct

www.cnn.com/ALLPOLITICS/	Minor Error	Correct
www.theaustralian.news.com.au	Correct	Correct
http://news.bbc.co.uk/2/hi/business	Major Error	Minor Error
http://news.bbc.co.uk/2/hi/asia-pacific	Minor Error	Minor Error (Error Rate Reduced)
http://www.reuters.com	Correct	Correct
http://news.ft.com (Financial Times)	Correct	Correct
http://dailytelegraph.news.com.au	Minor Error	Correct
www.iht.com (International Herald Tribune)	Correct	Correct
http://www.dailytimes.com.pk	Correct	Correct
http://news.xinhuanet.com/english	Correct	Correct
http://www.abc.net.au/news	Minor Error	Correct
http://news.ninemsn.com.au	Correct	Correct

Experiment results show that news articles were mostly extracted properly except BBC News (UK). After analyzing the web pages carefully, it was found that these web pages contained more than one content blocks in the table that also contains the news article, namely, the news article only occupies one of the table cell. Therefore, more experiments were conducted on this web site by using multiple documents. Experiment results show that the accuracy rate have been increased dramatically. It is because that although the content block is not correctly classified by the first step, other content blocks in the table are also extracted, but these extra content blocks in the extracted data are removed by the filtering process at the second step.

As it is shown in Table 1, by using the dynamically generated filter, the extraction accuracy has been increased considerably. The experiment confirmed the approach, which assumes that the news article is contained in a table formatting structure, and the advertisements and other content block data are embedded in nested table structure within the news article table, works well. This layout method is commonly used in most of news web sites, which makes proposed algorithms and their implementation a practically valuable tools.

During the experiment, the threshold value for validation was set to 1. Different combinations of weighting values have been tested. Experiment results showed that the validation process is highly effective. Moreover, the experimental validation results are not sensitive to the choices of weighting values.

#### 4.2 Constructing a news data set

An experiment was conducted to build a news data set from keywords “Interest Rate”. As there are large amount of news on the internet, this experiment restricted the time frame to 1 week and news sources within Australia.

Domain thesauruses were constructed by using 500 articles from each category: World, National, Business, Science (Technology), Sport and Entertainment. After the domain thesauruses have been constructed, their data remain the same for the whole experiment.

Table 2 shows the keywords (phases) used for each new search and the number of articles retrieved. The keywords for the next search were extracted from the data in the data set that has been constructed so far instead of the data from the last search results. The search process stops when the initial keywords are no longer in the most frequent keyword list generated from the last search results.

Table 2. The keywords used for each search in a data set construction process.

Keywords (phases) Used for the Search	Number of Most Re- lated Articles Retrieved	Most Frequent Keywords
Interest Rate	23	interest rate, housing market, bank, price, bond, finance, loan ...
Housing Market	10	housing market, finance, interest rate, price, value, bank ...
Price, Bank	30	bank, price, interest rate, share, oil, economy, stock ...
Finance, Bond	12	Bond, finance, housing market, interest rate, investor, price ...
Share, Investor	27	Share, investor, housing market, bank, price, finance, value ...

In total, the news data set contains 102 news articles. Each article in the data set was manually examined. Their contents are all within the scope of “interest rate”. Once the data set is constructed, it will be further processed and used as the information source for the negotiation agent.

## 5 Bringing the News to the Negotiation Table

The above described tools can be used directly in the “semi-automatic” negotiation scenario (scenario “SA” in Figure 1). In this case, the information request can be initiated either by the human participant or by the negotiation agent. In both cases the keywords for initiating the news “hunt” can be extracted out of the negotiation utterances. In the case, when the negotiation agent requests the news, the keywords are filtered automatically from the dialogue and are passed to the news extraction bots (possibly with some weights based on the relative intensity with which they occur during the negotiation). In the “SA” scenario, the body of the article together with a date/time stamp and the source identifier is sufficient, as the information is assessed by the human player. An information table that contains the retrieved news text, validity of the data, most frequent keywords and other parameters is delivered to an information aggregation agent for further processing so that the information can be used by the negotiation agent efficiently. The detailed discussion of the automated utilisation of retrieved information is beyond the scope of this paper.

## 5 Conclusions and Future Work

The curious negotiator is the long term work in automated negotiation systems. It will blend ‘strategic negotiation sense’ with ‘strategic information sense’ as the negotiation unfolds. This requires a system capable of providing information to the “negotiation table”. The smart data mining systems that support the negotiation agents are expected to operate under time-constraints and over dynamically changing corpus of information. They will need to determine the sources of information, the confidence and validity of these sources and a way of combining extracted information (models).

In this paper, we presented a method to extract relevant news article from news web sites regardless of the format and layout of the source. The article’s logical structure is firstly identified by using the table tags in the tagged files. An internal dynamic filter is built to further clean up the data. Finally, a validation method is developed to validate the retrieved data. Experiment results confirm that the overall approach and the corresponding methodology and algorithms can be applied to most of the news web sites with reasonable accuracy.

In the case when a Web page is not partitioned by table tags, proposed method relies on the availability of a second document from the same web site. Although using table for page layout is the most popular method, other content partition methods should also be implemented in the system to improve the extraction accuracy.

Though developed for the curious negotiator, proposed methods can be applied for content extraction from tagged documents in mobile phone and PDA browsing area. Mobile phone and PDA have relatively slow internet access and small display area. Therefore, presented algorithms can be applied for automatic detection and display of articles from news web sites on such devices with improved efficiency and visual effect.

## References

- Simoff, S. J. and J. K. Debenham: Curious negotiator. Proceedings of The Int. Conference on Cooperative Information Agents, CIA-2002, Madrid, Spain, Springer, Heidelberg (2002).
- Gerding, E. H., D. D. B. van Bragt, et al.: Multi-issue negotiation processes by evolutionary simulation: validation and social extensions. Proceedings Workshop on Complex Behavior in Economics. Aix-en-Provence, France, (2000).
- Gomes, A. and P. Jehiel (Forthcoming): Dynamic process of social and economic interactions: on the persistence of inefficiencies. *Journal of Political Economy*.
- Kraus, S.: Strategic Negotiation in Multiagent Environments. Cambridge, MA, MIT Press (2001).
- Ströbel, M.: Design of Roles and Protocols for Electronic Negotiations. *Electronic Commerce Research Journal*, Special Issue on Market Design (2001).
- Milgrom, P. and R. A. Weber: Theory of Auctions with Competitive Bidding. *Econometrica*, **50**(5), (1982).
- Watkins, M.: Breakthrough Business Negotiation-A Toolbox for Managers, Jossey-Bass (2002).
- Hand, D., H. Mannila, et al.: Principles of Data Mining. Cambridge, MA, MIT Press (2001).
- Franz, M., A. Ittycheriah, et al.: First Story Detection: Combining Similarity and Novelty Based Approaches. In Topic Detection and Tracking Workshop Report, (2001)
- Simoff, S. J. and J. K. Debenham: Time-constrained support for decision-making in e-market environments. Proceedings of the 6th International Conference of The International Society for Decision Support Systems ISDSS'01, London, UK, (2001).
- Chidlovskii, B., J. Ragetti, et al.: Automatic wrapper generation for web search engines. Proceedings of the 1st International Conference on Web-Age Information Management WAIM'00, Springer. (2000).
- Freitag, D. and N. Kushmerick: Boosted wrapper induction. Proceedings of the 17th National Conference on Artificial Intelligence AAAI-2000. (2000).
- Kushmerick, N. and B. Grace: The wrapper induction environment. Workshop on Software Tools for Developing Agents, AAAI-98. (1998).
- Kushmerick, N.: Wrapper induction: Efficiency and expressiveness. *Artificial Intelligence* **118**(1-2): 15-68. (2000)

- Muslea, I., S. Minton, et al.: STALKER: Learning extraction rules for semistructured, Web-based information sources. Proceedings of AAAI-98 Workshop on AI and Information Integration, Menlo Park, CA, AAAI Press. (1998).
- Gao, X. and L. Sterling: Semi-structured Data Extraction from Heterogeneous Sources. In T. Bratjevik D. Schwartz, M. Divitini, editor, Internet-based Knowledge Management and Organizational Memories, pages 83--102. Idea Group Publishing. (2000).
- McKeown, K. R., R. Barzilay, et al.: Columbia multi-document summarization: Approach and evaluation. Proceedings of the Workshop on Text Summarization, ACM SIGIR Conference, DARPA/NIST Document Understanding Conferences (DUC). (2001).
- Lin, S. H. and J. M. Ho: Discovering informative content blocks from Web documents. Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD2002, ACM Press. (2002).
- Salton, G.: Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer. Addison-Wesley, (1989).
- Hulth, A., J. Karlgren, A. Jonsson, H. Boström and L. Asker: Automatic Keyword Extraction Using Domain Knowledge, Proceedings of Second International Conference on Computational Linguistics and Intelligent Text Processing. (*CICLing 2001*). Mexico City, February 2001. LNCS 2004, Springer.
- Andrade M, and A. Valencia: Automatic extraction of keywords from scientific text: application to the knowledge domain of protein families, *Bioinformatics* (14) 600-607, (1998).



# Visualisation and exploration of scientific data using graphs

Ben Raymond and Lee Belbin

Australian Government, Department of the Environment and Heritage  
Australian Antarctic Division  
Channel Highway, Kingston 7050 Australia  
`ben.raymond@aad.gov.au`

**Abstract.** We present a prototype application for graph-based data exploration and mining, with particular emphasis on scientific data. The application has a Flash-based graphical interface and uses semantic information from the data sources to keep this interface as intuitive as possible. Data can be accessed from local and remote databases and files. The user can generate a number of graphs that represent different views of the data. Graphs can be explored using an interactive visual browser, or graph-analytic algorithms. We demonstrate the approach using marine sediment data, and show that differences in benthic species compositions in two Antarctic bays are related to heavy metal contamination.

## 1 Introduction

Structured graphs have been recognised as an effective framework for scientific data mining — e.g. [1, 2]. A graph consists of a set of nodes connected by edges. In the simplest case, each node represents an entity of interest, and edges between nodes represent relationships between entities. Graphs thus provide a natural framework for investigating relational, spatial, temporal, and geometric data [2]. Graphs have also seen a recent explosion in popularity in science, as network structures have been found in a variety of fields, including social networks [3, 4], trophic webs [5], and the structures of chemical compounds [6–8]. Networks in these fields provide both a natural representation of data, as well as analytical tools that give insights not easily gained from other perspectives.

The Australian Antarctic Data Centre (AADC) sought a graph-based visualisation and exploration tool that could be used both as a component of in-house mining activities, as well as by clients undertaking scientific analyses.

The broad requirements of this tool were:

1. *Provide functionality to construct, view, and explore graph structures, and apply graph-theoretic algorithms.*
2. *Able to access and integrate data from a number of sources.* Data of interest typically fall into one of three categories:

- databases within the AADC (e.g. biodiversity, automatic weather stations, and state of the environment reporting databases). These databases are developed and maintained by the AADC, and so have a consistent structure and are directly accessible.
  - flat data files (including external remote sensed environmental data such as sea ice concentration [9], data collected and held by individual scientists, and data files held in the AADC that have not yet been migrated into actively-maintained databases).
  - web-accessible (external) databases. Several initiatives are under way that will enable scientists to share data across the web (e.g. GBIF [10]).
3. *Be web browser-based.* A browser-based solution would allow the tool to be integrated with the AADC's existing web pages, and thus allow clients to explore the data sets before downloading. It would also allow any bandwidth-intensive activities to be carried out at the server end, an important consideration for scientists on Antarctic bases wishing to use the tool.
  4. *Have an intuitive graphical interface* (suitable for a general audience) that would also provide sufficient flexibility for more advanced users (expected to be mostly internal scientists).
  5. *Integrated with the existing AADC database structure.* To allow the interface to be as simple as possible, we needed to make use of the existing data structures and environments in the AADC. For example, the AADC keeps a data dictionary, which provides limited semantic information about AADC data, including the measurement scale type (nominal, ordinal, interval, or ratio) of a variable. This information would allow the application to make informed processing decisions (such as which dissimilarity metric or measure of central tendency to use for a particular variable) and thus minimise the complexity of the interface.

Existing software that we were aware of met some but not all of these requirements. A summary of a selection of graph software is presented in Table 1 (an exhaustive review of all available graph software is beyond the scope of this paper). This paper describes a prototype tool that can be used to create and explore graph structures from a variety of data sources. The graphical interface has been written as a Flash application; the server-side code is written in ColdFusion (our primary application development environment). The interface can also accept text-based commands for users wishing additional flexibility.

## 2 Methods

The exploratory analysis process can be divided into three main stages — graph construction; visual, interactive exploration; and the application of specific analytical algorithms. In practice, these components would be used in an interactive, cyclical exploratory process. We discuss each of these aspects in turn.

**Table 1.** A functional summary of a selection of graph software. BG: the package provides functionality for constructing graphs from tabular or other data (manual graph construction excluded); DB,WS: direct access to data from databases/web services; L&D: provides tools for the layout and display of graphs; A: provides algorithms for the statistical analysis of graphs; Int.: interface type; BB: is web browser-based. <sup>†</sup>Small graphs only. <sup>‡</sup>Designed for large graphs. \*Limited functionality when run as an applet

Package	BG	DB	WS	L&D	A	Int.	BB	Summary
GGobi[28]	✓	✓	✗	✓ <sup>†</sup>	✗	GUI	✗	General data visualisation system with some graph capabilities
Zoomgraph[29]	✓	✓	✗	✓ <sup>‡</sup>	✓	Text	✓*	Zoomable viewer with database-driven back end
UCINET[31]	✓			✓	✓	GUI	✗	Popular social network analysis package
Pajek[30]	✗			✓ <sup>‡</sup>	✓	GUI	✗	Analysis and visualization of large networks
Tulip[34]	✗			✓ <sup>‡</sup>	✓	GUI	✗	Large graph layout and visualisation
LGL[35]	✗			✓ <sup>‡</sup>	✗	GUI	✓	Large graph layout
GraphViz [36]	✗			✓	✗	Text	✗	Popular layout package
SUBDUE[13]	✗			✗	✓	Text	✗	Subgraph analysis package

## 2.1 Graph construction

Currently, data can be accessed from one or more local or remote databases (local in this context means “within the AADC”) or user files. Accessing multiple data sources allows a user to integrate their data with other databases, but is predictably made difficult by heterogeneity across sources. We extract data from local databases using SQL statements; either directly or mediated by graphical widgets. Local files can be uploaded using http/get and are expected to be in comma-separated text format. Users are encouraged to use standardised column names (as defined by the AADC data dictionary), allowing the semantic advantages of the data dictionary to be realised for file data. Remote databases can be accessed using web services. Initially we have provided access only to GBIF data [10] through the DiGIR protocol. Data from web service sources are described by XML schema, which can be used in a similar manner to the data dictionary to provide limited semantic information.

To construct a graph representation of these data, the user must specify which variables are to be used to form the nodes, and a means of forming edges between nodes. Nodes are formed from the discrete values (or  $n$ -tuples) of one or more variables in the database. The graphical interface provides a list of available data sources, and once a data source is selected, a list of all variables provided by that data source. This information comes from the column names in a user file or database table, or from the “concepts” list of a DiGIR XML resource file. Available semantic information is used to decide how to discretise the node variables. Continuous variables need to be discretised to form individual nodes.

A simple equal-interval binning option is provided for this purpose. Categorical or ordinal (i.e. discrete) variables need no discretisation, and so this dialogue is not shown unless necessary.

Once defined, each node is assigned a set of attribute data. These data are potentially drawn from all other columns in the database. The graphical interface allows attribute data to be drawn from a different data source provided that the sources can be joined using a single variable. More complex joins can be achieved using text commands. Attribute data are used to create the connectivity of the graph. Nodes that share attribute values are connected by edges, which are optionally weighted to reflect the strength of the linkage between the nodes. The application automatically chooses a weighting scheme that is appropriate to the attribute data type; this choice can be overridden by the user if desired.

Once data sources and variables have been defined, the application parses the node attributes to create edges, and builds an XML (in fact GXL, [11]) document that describes the graph. The graph can be either visually explored, or processed with one of many graph-based analytic algorithms.

## 2.2 Graph visualisation

Graph structures are displayed to the user in an interactive graph browser. The browser is a modified version of the Touchgraph LinkBrowser [12], which is an open-source Java tool for graph layout and interaction. Layout is accomplished using a spring-model method, in which each edge is considered to be a spring, and the node positions are chosen to minimise the global energy of the spring system. Nodes also have mutual repulsion in order to avoid overlap in the layout.

While small graphs can reasonably be displayed in their entirety, large graphs often cannot be displayed in a comprehensible form on limited screen real estate. We solve this problem by allowing large graphs to be explored as a dynamic series of smaller graphs (see below). We discuss alternative approaches, such as hierarchical views with varying level of detail, in the discussion.

Interaction with the user is achieved through three main processes: node selection, neighbourhood adjustment, and edge manipulation. The displayed graph is focused on a selected node. The neighbourhood setting determines how much of the surrounding graph is displayed at any one time. This mechanism allows local regions of a graph to be displayed. Edge manipulation can be done using a slider that sets the weight threshold below which edges are not displayed. It is difficult to judge *a priori* which edges to filter out, as weak edges can obscure the graph structure in some cases but may be crucial in others. A practical solution is to create a graph with relatively high connectivity (many weak links), and then allow the user to remove links in an interactive manner.

The graph layout is done dynamically, and changes smoothly as the user varies the interactive settings. The graph layout uses various visual properties of the nodes and edges to convey information, including colour, shape, label, and mouse-over popup windows. We also allow attributes of the nodes to set the graph layout. This is particularly useful with spatial and temporal data.

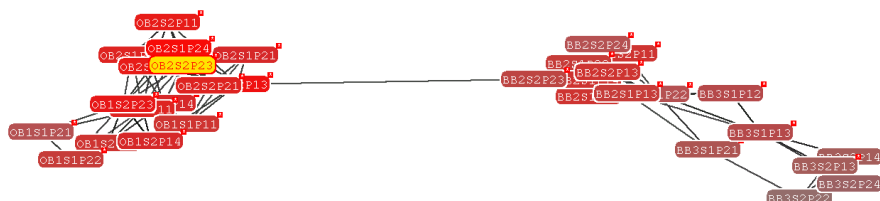
An alternative visualisation option is to save the XML document and import it into the user's preferred graph software. This might be appropriate with extremely large graphs, since this visualisation tool does not work well with such graphs.

### 2.3 Analytical tools

The fields of graph theory and data mining have developed a range of algorithms that assess specific properties of graph structures, including subgraph analyses (e.g. [13–17]), connectivity and flow [8], graph simplification [5, 18], clustering, and outlier detection [19, 20]. Many of the properties assessed by these tools have interpretations in terms of real-world phenomena (e.g. [21–23]) that are not easily assessed from non-graph representations of the data. These provide useful analytical information to complement existing scientific analyses, and also the possibility of building graphs based on analyses of other graphs.

A simple but very useful example is an operator that allows the similarity between two graphs to be calculated. We use an edge-matching metric, equal to the number of edges that appear in both graphs, as a fraction of the total number of unique edges in the two graphs (an edge is considered to appear in both graphs if the same two nodes appear in both graphs, and they are joined by an edge in both graphs). This provides a simple method for exploring the relationships between graphs, and also a mechanism for creating graphs of graphs: given a set of graphs, one can construct another graph  $\mathcal{G}$  in which each graph in the set is represented by a node. Using a graph similarity operator, one can calculate the similarity between each pair of graphs in the set, and use this similarity information to create weighted edges between the nodes in  $\mathcal{G}$ . The visualisation tool allows a node in a graph to be hyperlinked to another graph, so that each node in a graph of graphs can be explored in its own right. We demonstrate these ideas in the Results section, below.

We have chosen not to implement other algorithms at this stage, concentrating instead on the graph construction and visual exploratory aspects. We raise future algorithm development options in the Discussion section, below.



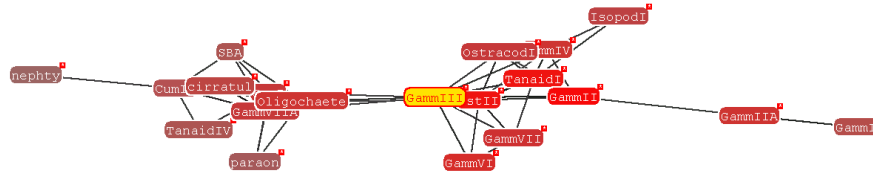
**Fig. 1.** A graph of Antarctic marine sample sites, linked by their species attribute data. Sites are clearly separated into two clusters on the basis of their species, indicating two distinct types of species assemblage. Node labels are of the form  $XYsSpPr$  and denote the position of the sample in the nested experimental hierarchy.  $BBY$  denotes samples from one of two locations in contaminated Brown Bay and  $OBY$  denotes uncontaminated O'Brien Bay;  $s$  denotes the site number within location;  $p$  denotes the plot number within site; and  $r$  denotes the core replicate number within plot

### 3 Results

We use a small Antarctic data set to demonstrate the graph construction and visualisation tools in the context of an exploratory scientific investigation.

Australia has an on-going research programme into the environmental impacts of human occupation in Antarctica (see <http://www.aad.gov.au/default.asp?casid=13955>). A recent component of this programme was an investigation into the relationships between benthic species assemblages and pollution near Australia's Casey station [24]. Marine sediment samples were collected from two sites in Brown Bay, which is adjacent to a disused rubbish tip and is known to have high levels of many contaminants. Samples were collected at approximately 30 m and 150 m from the tip. Control samples were collected from two sites in nearby, uncontaminated O'Brien Bay. Four replicate samples were collected from two plots at each site, giving a total of 32 samples. Sediment samples were collected by divers using plastic corers and analysed for fauna (generally identified to species or genus level) and heavy metal concentrations (Pb, Cd, Zn, As, Cr, Cu, Fe, Ni, Ag, Sn, Sb). These metals are found in man-made products (e.g. batteries and steel alloys) and can be used as indicators of anthropogenic contamination. Details of the experimental methods are given in [24].

This data set has a very simple structure, comprising a total of 14 variables: `site_name`, `species_id`, `species_abundance`, and measured concentrations of the 11 metals listed above. Site latitude and longitude were not recorded but the `site_name` string provides information to the site/plot/replicate level (see Fig. 1 caption). All of the above information appears in one database table. The `species_id` identifier links to the AADC's central biodiversity database, which provides additional information about each species (although we do not use this additional information in the example presented here). Standard practice would



**Fig. 2.** A graph of Antarctic marine species, linked by their site attribute data. This graph provides complementary information to that shown in Fig. 1, and confirms that two distinct species assemblages exist, with possible outliers *GammIIA*, *GammI*, and *nepty*. Eight other disconnected outlier species are not shown

normally also see a separate table for the sample site details, but in this case there are only a small number of sample sites that are specific to this data set.

Despite the simplicity of the data set, there are a large number of graphs that can be generated. The key questions to be answered during the original investigation related to spatial patterns in species assemblages, and the relationships of any such patterns to contamination (heavy metal concentrations).

Spatial patterns in species assemblages can be explored using sites as nodes, and edges generated on the basis of species attribute data. To create this graph, we needed only to select `site_name` as entities, and `species_id` as attributes in the graphical interface. Both of these variables were recognised by the data dictionary as categorical, and so no discretisation was needed. Furthermore, an edge weighting function suitable for species data was automatically selected. This function is based on the Bray-Curtis dissimilarity, which is commonly used with ecological data:

$$w_{ij} = 1 - \sum_k \frac{|x_{ik} - x_{jk}|}{(x_{ik} + x_{jk})}, \quad (1)$$

where  $w_{ij}$  denotes the weight of the edge from node  $i$  to node  $j$ , and  $x_{ik}$  denotes the  $k$ th attribute of node  $i$ .

The resultant graph is shown in Fig. 1. Weak edges have been pruned, leaving a core structure of two distinct clusters of sites: the left-hand cluster corresponds to sites from O’Brien Bay; the right-hand cluster Brown Bay. This strong clustering suggests that the species assemblages of the two bays are distinct. Furthermore, each cluster shows spatial autocorrelation — that is, samples from a given site in a given bay are most similar to other samples from the same site (e.g. BB3 nodes are generally linked to other BB3 nodes). We generated an alternative view of the data by swapping the definitions for entity and attribute, giving a graph of **species\_id** nodes with edges calculated on the basis of **site\_id** attribute data (Fig. 2). This graph confirms the presence of two broadly distinct species assemblages, but suggests that there are several outlier species that may



**Fig. 3.** The same graph as Fig. 1, but with edge colouring changes to indicate similarity of chromium between sites. Darker edges are those that are better “explained” by chromium patterns (see text for details). The O’Brien Bay cluster (left) has a strong similarity of chromium values within it, whereas the Brown Bay cluster has dissimilar values both within itself and to the O’Brien Bay cluster. These results, and similar results with other metal variables, suggest that species differences between the two bays may be related to heavy metal concentrations

diverge from this bimodal pattern of spatial distribution.

Having established some patterns in species assemblages, we wish to explore the relationships between these patterns and measured metal contamination. A convenient method for this is through the graph similarity operator. We generated a second graph of sites, using chromium as attribute data (graph not shown), and made an edge-wise weight comparison between the site-species graph and the site-chromium graph. The result is shown in Fig. 3. The structure of this graph is identical to that in Fig. 1, but the colouring of the edges indicates the weight similarity. Darker grey indicates edges that have similar weights in both the site-species and site-chromium graphs. Samples from the uncontaminated O’Brien Bay have similar chromium values (in fact, mostly near zero). More notably, the single edge linking the O’Brien Bay cluster to the Brown Bay cluster is not well explained in terms of chromium, suggesting that the clustering might be related to differences in chromium values. (The general dissimilarity between chromium values in the Brown Bay cluster is an artefact of their high but variable values, which have fallen into different bins during the discretisation process. We discuss the problem of discretising attributes below). Similar results were obtained using the other metal variables, supporting the notion that the benthic species assemblages of these bays is related to heavy metal contamination.

Finally, we use a graph of graphs to explore the similarities between the spatial patterns of the various heavy metals. We generated 11 graphs, one for each metal, using sites as entities and the metal as attribute data. The pairwise similarities between each of these graphs were calculated. Fig. 4 shows the resultant graph, in which each node represents an entire site-metal graph, and the edges indicate the similarities between those graphs. The graph suggests that copper, lead, iron, and tin are distributed similarly, and that their distribution is different to that of nickel, chromium, and the other metals. This was confirmed by inspecting histograms of metal values at each location: values of copper, lead, iron, and tin were higher at one of the Brown Bay locations (the one closest to



**Fig. 4.** A graph of graphs. Each node represents an entire subgraph — in this case, a graph of sites linked by a metal attribute. This graph of graphs indicates that the spatial distributions of copper, lead, iron, and tin are similar, and different to those of nickel, chromium, and the other metals

the tip) than the other, whereas the remaining metals showed similar levels at each of the two Brown Bay locations.

## 4 Discussion

Graphs have been previously been recognised for their value in data mining and exploratory analyses. However, existing software tools for such analyses (that we were aware of) did not meet our requirements. We have outlined a prototype web-based tool that builds graph structures from data contained in databases or files, and presents the graphs for visual exploration or algorithmic analysis.

The construction phase requires the user to define the variables that will be used to form the graph nodes. While there may be certain definitions that are logical or intuitive in the context of a particular database (for example, it is probably intuitive to think of species as nodes when exploring a database of wildlife observations), the nodes can in fact be an arbitrary combination of any of the available variables. This is a powerful avenue for interaction and flexibility, as allows the user to interpret the data from a variety of viewpoints, a key to successful data mining [25].

One of the notable limitations of our current implementation is the requirement that attribute data be discrete. (Edges are only formed between nodes that have an exact match in one or more attributes). Continuous attributes must be discretised, which is both wasteful of information and can lead to different graph structures with different choices of discretisation method. Discretisation is potentially particularly problematic for Antarctic scientific data sets, which tend not only to be relatively small but also sparse. Sparsity will lead to few exact matches in discretised data, and to graphs that may have too few edges to convey useful information. Future development will therefore focus on continuous attribute data.

The visualisation tool that we have discussed is best suited to relatively small graphs. This is generally not an issue with Antarctic scientific data sets, which tend to be of manageable size, but conventional data mining of very large data sets would be problematic. Other visualisation tools, specifically designed for large graphs (e.g. [18, 26, 27]) might be useful for visualising such graphs. FADE [18] and MGv [26] use hierarchical views that can range from global structure of a graph with little local detail, through to local views with full detail.

Many other packages for graph-based data exploration exist, and we have incorporated the features of some of these into our design. The GGobi package [28] has a plugin that allows users to work directly with databases. GGobi also ties into the open-source statistical package R to provide graph algorithms. Zoomgraph [29] takes the same approach. This is one method of providing graph algorithms without the cost of re-implementation. Another is simply to pass the graph to the user, who can then use one of the many freely-available graph software packages (e.g. [30–33]). Yet another approach, which we are currently investigating, is the use of analytical web services. Our development has been done in Coldfusion, which can make use of Java and can also expose any function as a web service. This may allow us to deploy an existing Java graph library such as Jung [33] as a set of web services. This approach would have the advantage that external users could also make use of the algorithms, by passing their GXL files via web service calls.

## References

1. T. Washio and H. Motoda, *State of the art graph-based data mining*, SIGKDD Explorations: Newsletter of the ACM Special Interest Group on Knowledge Discovery & Data Mining, 5(1) (2003), pp. 59–68
2. M. Kuramochi, M. Desphande, and G. Karypis, *Mining Scientific Datasets Using Graphs*, in Next Generation Data Mining, H. Kargupta, A. Joshi, K. Sivakumar, and Y. Yesha (eds), MIT Press (2003)
3. R.L. Brieger, *The analysis of social networks*, in Handbook of Data Analysis, M. Hardy and A. Bryman (eds), London, SAGE Publications (2004), pp. 505–526
4. D. Lusseau and M.E.J. Newman, *Identifying the role that individual animals play in their social network*, Biology Letters, in press
5. J.J. Luczkovich, S.P. Borgatti, J.C. Johnson, and M.G. Everett, *Defining and measuring trophic role similarity in food webs using regular equivalence*, Journal of Theoretical Biology, 220(3) (2003), pp. 303–321
6. S.-H. Yook, Z.N. Oltavai, and A.-L. Barabási, *Functional and topological characterization of protein interaction networks*, Proteomics, 4 (2004), pp. 928–942
7. J. Gonzalez, L. B. Holder, and D. J. Cook, *Application of graph-based concept learning to the predictive toxicology domain*, in Proceedings of the Predictive Toxicology Challenge Workshop (2001)
8. L. De Raedt and S. Kramer, *The level wise version space algorithm and its application to molecular fragment finding*, in Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (2001)

9. J. Comiso, *Bootstrap sea ice concentrations for NIMBUS-7 SMMR and DMSP SSM/I*, Boulder, CO, USA: National Snow and Ice Data Center (1999, updated 2002)
10. Global Biodiversity Information Facility, <http://www.gbif.net>
11. A. Winter, B. Kullbach, and V. Riediger, *An overview of the GXL graph exchange language*, Software Visualization, S. Diehl (ed.), Springer-Verlag (2001)
12. A. Shapiro, *Touchgraph*, <http://www.touchgraph.com>.
13. D.J. Cook and L.B. Holder, *Graph-based data mining*, IEEE Intelligent Systems, 15(2) (2000), pp. 32–41
14. M. Kuramochi and G. Karypis, *Finding frequent patterns in a large sparse graph*, in Proceedings of the SIAM International Conference on Data Mining, Florida (2004)
15. C. Cortes, D. Pregibon, and C. Volinsky, *Computational methods for dynamic graphs*, J. Computational and Graphical Statistics, 12 (2003), pp. 950–970
16. A. Inokuchi, T. Washio, and H. Motoda, *Complete mining of frequent patterns from graphs: mining graph data*, Machine Learning, 50 (2003), pp. 321–354
17. X. Yan and J. Han, *CloseGraph: Mining closed frequent graph patterns*, in Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2003)
18. A. Quigley and P. Eades, *FADE: graph drawing, clustering, and visual abstraction*, Proceedings of the 8th International Symposium on Graph Drawing (2000), pp. 197–210
19. S. Shekhar, C.-T. Lu, p. Zhang, *Detecting graph-based spatial outliers: algorithms and applications (a summary of results)*, in Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2001), pp. 371–376
20. C.C. Noble and D.J. Cook, *Graph-based anomaly detection*, in Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2003), pp. 631–636
21. M. Girvan and M.E.J. Newman, *Community structure in social and biological networks*, Proc. Natl. Acad. Sci. USA 99 (2002), pp. 7821–7826
22. B. Drossel, A.J. McKane, *Modelling food webs*, in Handbook of Graphs and Networks: From the Genome to the Internet, S. Bornholdt and H.G. Schuster (eds), Wiley-VCH, Berlin (2003)
23. J. Moody, *Peer influence groups: identifying dense clusters in large networks*, Social Networks, 23 (2001), pp. 216–283
24. J.S. Stark, M.J. Riddle, I. Snape, R.C. Scouller, *Human impacts in Antarctic marine soft-sediment assemblages: correlations between multivariate biological patterns and environmental variables at Casey Station*, Estuarine, Coastal and Shelf Science, 56 (2003), pp. 717–734
25. J. Neville and D. Jensen, *Supporting relational knowledge discovery: lessons in architecture and algorithm design*, Proceedings of the International Conference on Machine Learning Workshop on Data Mining Lessons Learned (2002)
26. J. Abello and J. Korn, *MGV: a system for visualizing massive multi-digraphs*, IEEE Transactions on Visualization and Computer Graphics, 8 (2002), pp. 21–38
27. G.J. Wills, *NicheWorks — interactive visualization of very large graphs*, J. Computational and Graphical Statistics, 8(2) (1999), pp. 190–212
28. D.F. Swayne, A. Buja, and D. Temple Lang, *Exploratory visual analysis of graphs in GGobi*, in Proceedings of the 3rd International Workshop on Distributed Statistical Computing, Vienna, 2003
29. E. Adar and J.R. Tyler, *Zoomgraph*, <http://www.hpl.hp.com/research/idl/projects/graphs/>

30. V. Batagelj and A. Mrvar, *Pajek - Program for Large Network Analysis*, <http://vlado.fmf.uni-lj.si/pub/networks/pajek/>
31. S. Borgatti and R. Chase, *UCINET: social network analysis software*, <http://www.analytictech.com/ucinet.htm>
32. B. Bongiovanni, S. Choplin, J.F. Lalande, M. Syska, and Y. Verhoeven, *Mascotte Optimization project*, <http://www-sop.inria.fr/mascotte/mascotpt/index.html>
33. S. White, J. O'Madadhain, D. Fisher, Y.-B. Boey, *Java Universal Network/Graph Framework*, <http://jung.sourceforge.net>
34. D. Auber, *Tulip — A Huge Graph Visualization Framework*, <http://www.tulip-software.org/>
35. A.T. Adai, S.V. Date, S. Wieland, and E.M. Marcotte, *LGL: creating a map of protein function with an algorithm for visualizing very large biological networks*, *Journal of Molecular Biology*, 340 (1) (2004), pp. 179–190
36. J. Ellson and S. North, *Graphviz - Graph Visualization Software*, <http://www.graphviz.org/>

# **An evaluation of the utility of two data mining project methodologies**

Marcel van Rooyen

This research was done in the E-Markets Group, Institute of Information and Communication Technologies, University of Technology Sydney, Australia.

Fax/phone: +61-2-9876-6006

[marcel@it.uts.edu.au](mailto:marcel@it.uts.edu.au)

**Abstract.** This paper aims to stimulate debate on a subject, which is rarely published about in the data mining and business domains. In it, we critically evaluate the project management utility of CRISP-DM and Data Mining Projects Methodology (DMPM) of SAS Institute, in a business decision-support environment. We describe the limited utility of both methodologies in a number of dimensions. The paper finally announces a research project which is creating a data mining project methodology, which offers improved project management utility.

*Key words: Business value, decision-making, solution execution, project management utility, Data Mining Projects Methodology, CRISP-DM, information discovery, knowledge development, concept drift.*

## **Introduction**

Data mining is a fusion of techniques from a number of specialised and diverse domains e.g. data management, machine learning, statistics, computing, visualisation, knowledge acquisition etc. (Han and Kamber 2001). Some business domain understanding and knowledge complement the expertise of the data miner in these domains.

Business decision making is a fusion of principles and practise from the specialised and diverse domains of Information and Knowledge Management, Strategic Management, Cognitive Psychology, Operations Management, Project Management, Technology Management, Change Management, Risk Management, and business domain knowledge. The expertise of the business decision-maker in these domains may be complemented by some understanding and knowledge about data mining technology.

The business decision-making process essentially is a knowledge developing business process, with a sequential application of interpretive, comparative, and decisive activities. The purpose of business knowledge development is developing executable solutions for business problems or business opportunities. This knowledge development process progressively builds knowledge from information.

Human cognition is the main tool in this knowledge developing process. However, the business decision-maker may depend on a supporting technology to provide informational input during any activity in this process. The potential for data mining

to be used as a tool for supporting the business decision-making process has been recognised for a number of years (Ganti, Gehrke et al. 1999; Hastie, Tibshirani et al. 2001). The utility of the data mining tool set, is its ability to discover information, which was not discoverable using traditional analytics tools (Han and Kamber 2001, pp. xix, 4).

Despite the potential for the output of data mining to support business decision-making, data mining practitioners are still finding resistance to the uptake of data mining by the business community. Where it is taken up, data mining practitioners sometimes experience the non-utilisation of the data mining output by the business decision-maker for solution design. In worst cases, the business may base the design of a solution on inexecutable output of data mining, resulting in the failure of the business solution. Industry leaders (Fayyad 2004) (Pyle 2004, Rule 8) have identified the main cause of resistance to data mining and of its failure sometimes, as a communication gap between data mining experts and business domain experts. We list Pyle's rules in the Appendix.

The communication problem is rooted in the different skills bases of the two communities, and the different nature of their tools of trade. At best, the only overlap between the data miner and the business decision-maker's skill sets, is their reciprocal non-expert understanding of each other's domains and tools.

A further cause identified in the literature for the non-uptake of data mining, the non-use of its results, and failing business solutions, is a preoccupied focus on the technology itself, instead of focusing on how the technology should support the business's needs (Pyle 2004, Rules 3, 5).

About four years ago, the data mining industry recognised these problems and their causes, and developed two major data mining project methodologies. They are Data Mining Projects Methodology (DMPM) (SAS Institute 2000), and CRISP-DM (Chapman, Clinton et al. 1999-2000).

Data mining methodologies propose project management utility (Pyle 1999, p.10). (SAS Institute 2000, p.xi) (Chapman, Clinton et al. 1999-2000, p.3). Their purpose is to:

- bridge the gap between the data mining and business communities; and to
- provide practical guidance about key project decisions where data mining is used for business decision-support.

In this paper, a corporate business manager, who is qualifying to become a data mining engineer, gives his views on the project management utility of these two methodologies, in the business decision-supporting environment. We present the views under seven headings.

## 1 CRISP-DM: Distinction between Information and Knowledge

We present the following practical distinctions between data, information, and knowledge:

1. *Data* are the non-mentally recorded measurements about events, their connections and relationships (Pyle 1999, p.2). The *origin* of data is measurement and recording. An example of a tool for the recording of data, is a data base;

2. *Information* is an entity which reduces uncertainty about a state or event (Lucas 2000, p.26), and as such is a *measurable signal* that is either present or absent (Pyle 1999, pp.406ff.). The presence or absence of information, means that its origin is *discovery*. Data mining is a tool for discovering patterns of information (Ganti, Gehrke et al. 1999) Despite the common use of the term *Knowledge Discovery in Databases* (KDD) in the data mining literature e.g. (Han and Kamber 2001, pp.5ff.), there is sufficient evidence in the data mining literature itself, to say that what data mining discovers, resides at the level of information (Hastie, Tibshirani et al. 2001) (Pyle 1999, p.23 and Chapter 11). DMPM recognises this too (SAS Institute 2000);
3. information on its own does not have executability, and without executability, it does not have business value (Meltzer 2000, p.1). *Knowledge* adds to the informational signal the insights and understanding about the causes and implications of the information (Kogut and Zander 1992). Knowledge is the executable *know-how* of the information – the *which actions produce which results, and how and when to take them*. (Zikmund 2003, p.21) (Pyle 1999, p.2). Developing knowledge from relevant information, constitutes the business value of that information (Grant 1996, p.375). In business, executable knowledge constitutes a solution for a problem or an opportunity;
4. this invention or development of knowledge, is the result of human cognition (Gibson, Ivanchevich et al. 1991; Schön 1995; Lucas 2000, pp.26ff.). In business, this cognition takes place formally in the knowledge developing business activities, which we described in the Introduction to this paper.

CRISP-DM's vocabulary is that data mining *discovers knowledge*. This is contra the above-said four concepts, and an oxymoron. When used in a data mining project methodology, such terminology indicates that the above distinctions are not supported by the project methodology. A project methodology which does not maintain the distinction between information and knowledge will be strained to offer project management utility in the business decision-support environment.

A further result from such terminology is that it may lead to a perception of a professional threat among the less informed business community. The perceived threat is from a technology which develops knowledge, potentially replacing their professional cognitive skills.

In both cases, the gap between the data mining and business communities has been perpetuated, by the very methodology, which is supposed to reduce the gap.

DMPM maintains the distinction between information discovery and knowledge development.

## 2 Diagnostic Technique for Defining Project Goals

Business decision-makers nuance their use of decision supporting technologies. The nuancing depends on their perceptions about a business problem or opportunity, their goal with the problem or opportunity, and how they think the technology may support that goal. A practically way of nuancing the use of decision-support technology is:

1. the use of technology to discover information, which confirms or denies the existence of a potential problem or opportunity – the decision-maker suspects that there is a problem or an opportunity, and the goal is to seek information with which to confirm that suspicion. Confirmation of the suspicion is followed by;
2. the use of technology to discover information about the main components of the problem. The goal of the decision-maker with the information is to infer from it an understanding about the nature of the problem or opportunity (Hastie, Tibshirani et al. 2001, p.99). Following successful understanding, the decision-maker cognitively formulates a hypothesis - or hypotheses - about solving the problem or about realising the opportunity (Schön 1995). Successful hypothesis formulation is followed by;
3. the use of technology with the goal of discovering information, which the decision-maker develops into knowledge, which supports or refutes the hypotheses about the solution or response. A supported hypothesis is followed by;
4. the use of technology, for discovering information, with the goal of cognitively developing the supported hypothesis into executable knowledge, or a business solution; for which
5. the use of technology with the goal to support the execution of the solution.

Aligning the *project's goals* with the business's goals about a problem or opportunity is a necessary utility requirement for a decision-support project methodology. In complex BI situations, there can be multiple business goals – reflecting the above nuancing – which could result in a complex project goal structure, and a chain of supporting models (Wedel and Kamakura 2000, p.245).

Both methodologies recognise the need for the *project* to produce results which best support the business's goals (SAS Institute 2000, pp. 50, 24, 26, 111, 143) (Chapman, Clinton et al. 1999-2000, Business understanding section). Both methodologies have activity sets which formulate these business goals. In DMPM, it is *Define the Business Problem*, and in CRISP-DM, it is *Business understanding*.

Both methodologies also distinguish between the use of data mining for supporting business decision-making, and the use of data mining as a tool for executing the business solution. This is apparent from CRISP-DM's *Deployment* section (Chapman, Clinton et al. 1999-2000, pp.60ff.), and DMPM's *Implement in production* section (SAS Institute 2000, pp.71ff.).

Despite these provisions in both methodologies, our view is that neither has sufficient content about formulating the *problem with the problem*, and about formulating the *what* which is being pursued by the project (Pyle 2004, Rule 1). In project management terminology, both methodologies lack diagnostic technique for formulating the decision-support goals about the business problem or opportunity, and for converting those into project goals (Chatfield and Johnson 2000, p.2) (Westphal and Blaxton 1998, Chapter 3) (Pyle 1999, pp.12-21).

There are three unwanted results from this limitation. It does not communicate to the business community, the versatility of data mining in business decision-support across a spectrum of goals. The above five points is an example of a possible goal spectrum. This non-communication detracts from the methodologies' utility as a communication instrument, and as a project methodology. The author believes that

this is one of the causes for the low uptake rate of the technology by business decision-makers.

Second, where a business user does accept the methodologies' project management utility at face value, it may result in the *unawares* ill-definition of a data mining project. This may be caused by an insufficient definition of the business goals about the problem, or by an insufficient definition of the supporting project goals. Such a project may be subsequently abandoned during the project evaluation phase, when deficiency is discovered in the results of the modeling. In fact, CRISP-DM actually makes provision for the abandonment of a project for this very reason (Chapman, Clinton et al. 1999-2000, p.57)!

Three, where an organisation does not detect this ill-definition during an evaluation, and executes a business solution based on the output from the data mining, such a business solution is bound to fail. The fall-out from such failure, almost certainly will not distinguish between the data mining technology and its supporting project methodology, and will negatively affect the acceptance of data mining *as a technology*.

### 3 Bridging Technique Linking Data Mining Plan to Project Goals

This point is about establishing a data mining plan, which supports the project's goals. Technically, this section is about the absence of bridging techniques, for establishing a sufficient link between the data mining plan and the knowledge developing activities. Such links assure the formulation of a data mining plan, which *directly* supports the *project's goals*, and *indirectly* supports the *business goals* (Cooley 2003, p.608).

Such links should be established at each iteration within the project. In the event that a data mining project methodology comes into existence, which embeds multiple knowledge developing activities among their supporting data mining activities, there should be links at each interface of knowledge developing and information discovery activities. Such a methodology would greatly enhance its project management utility, if bridging techniques were offered for establishing the link.

In the case of CRISP-DM and DMPM as they stand, the link would apply between the formulation of the data mining plan and DMPM's *Define the Business Problem*, and in CRISP-DM's *Business understanding*.

We demonstrate the concept with an example from the CRM software industry. A CRM software vendor has a software product, and a project methodology for its implementation. The methodology purports to assure that once the CRM system is implemented, it will optimally support the business's particular needs. The astute potential business client - who has already had a disappointing experience with the implementation of an ERP system - is cautious about the vendor's project methodology assuring such a desirable result from a technology implementation. That potential client then asks the CRM vendor a question, possibly in a number of ways: what *technique* do you use in your project methodology, or what *project activities* do you have, to assure that the CRM software is best configured for our needs? *How* do you formulate the technical *objectives and plan* for software configuration? *What* is it

in our business goals, upon which you base the plan with the technology? The uptake and application – or not - of the CRM software by the client, will be quite dependent on satisfactorily answering these questions. The commercial uptake of data mining similarly is dependent on satisfactorily answering the same questions about its project methodologies.

We systematically researched both methodologies for what could be termed a description of bridging technique. The search was for the use of the words *objective*, *goals*, *plan*, and *strateg\**, within the context of the data mining *itself*, and not the project. The words *objective*, *goal*, *plan* and *strategy* are key words within the knowledge developing business activities (Pearce and Robinson 1991) (Kotler 1988).

CRISP-DM (Chapman, Clinton et al. 1999-2000, Business understanding, subtask Determine data mining goals) bases the link on the *business objectives/goals* (pp. 17, 18, 40), and the *business success criteria* (pp.17, 36, 69). CRISP-DM's *bridging technique* is translating these *business objectives/goals* into technical data mining *goals* (p.40). Our research results on telecommunication data, which are nearing publication, find this technique limiting, and finds any link to *business success criteria* unsuitable.

In our interpretation of the results from the same search in DMPM (SAS Institute 2000), we found no certain basis for a view, that the methodology links the data mining *objective(s) / goals / plans / strateg\** to the understanding about the *objective(s) / goals / plans / strateg\** of the business problem. The word *plan* is used once about the *modeling* plan (p.143) in a checklist, but what the plan's *base* is, is not given. The word *strateg\** is not used once with reference to data mining or modeling. There is a data mining *approach* (p.50) but what its *objective(s) / goals / plans / strateg\** are, is not specified. DMPM assesses the model against business *objectives* (p.143), but there is nothing to suggest a *linking* with the data mining's *objective(s) / goals / plans / strateg\** at the outset. DMPM applies data mining techniques to established business *objectives* (p.29); this gives a *base*, but the *link* to the model's *objective(s) / goals / plans / strateg\** is not established.

We could not find anything in the methodologies, which in our practical view can be considered a description of bridging technique. We acknowledge that the two project methodologies recognise the *importance* of the link, and even have the *intention* to establish such links, but such recognition is insufficient in practice (Pyle 2004, Rule 7). This detracts from their decision-support project management utility.

This deficiency perpetuates perceptions within the sceptical factions of the business community, that data mining is too complex to be linked to the business's needs. In the event where a business uses one of the methodologies, it may result in the ill-definition of a data mining plan, with consequently disappointing results.

#### 4 Knowledge Developing Activities

The success of any project, which combines decision-making with a supporting technology, depends on the execution of a process. This process consists of an interwoven chain of technical and cognitive activities. The purpose of that project is

to generate executable knowledge, and therefore a business solution for the problem or opportunity. We believe that this condition also applies to data mining projects.

In the triangle in Figure 1 below, we visually express a practical view about the knowledge outputs, which are required from the project, for producing an executable business solution. These outputs are sequenced bottom-to-top. They incrementally add business value during the project, through generating knowledge about the project goals. The outputs dovetail with the problem nuancing we offered in section two. These knowledge outputs, and their added business value, are:

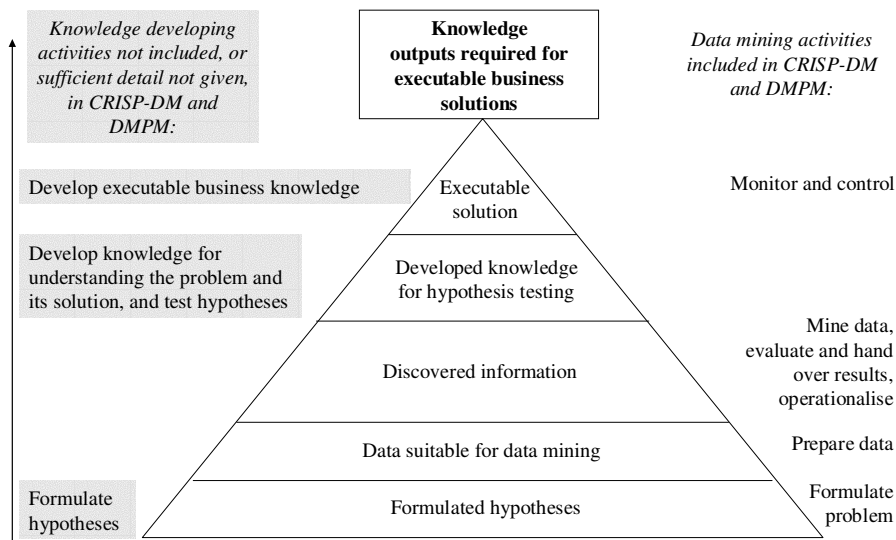
- formulated hypotheses about the nature of the problem or opportunity and its solution;
- knowledge about which data is relevant to mining about the problem, and about the signal it contains after we have prepared it;
- discovered information, which will be developed into knowledge for hypothesis testing;
- developed knowledge for hypotheses testing;
- developed knowledge, which constitutes the executable solution for the problem or solution.

In the previous section, we saw that both methodologies start with data mining activities, which develop knowledge about the business's goals with the problem. Those data mining activities are represented in the right-hand column of Figure 1, by the data mining activity *Formulate problem*. We express our view, that the *Formulate problem* activities of both methodologies have insufficient content about developing knowledge in the form of hypotheses for problem understanding and solution development. We express that deficiency in the left-hand column of Figure 1, with the greyed rectangle containing *Formulate hypotheses*.

Both methodologies then develop a sequence of data mining task sets based on the output of *Formulate problem*. These activities and their sequence are the remaining data mining activities in the right-hand column of Figure 1. These data mining activities also include evaluation activities for the discovered information (Chapman, Clinton et al. 1999-2000, Evaluation) (SAS Institute 2000, Assess). Both methodologies terminate with the handing over of business reports, the operationalising of the models, and activities which express the need for developing monitoring and maintenance plans (Chapman, Clinton et al. 1999-2000, Deployment section) (SAS Institute 2000, Implement in Production and Review sections).

We express our view that the information evaluating activities in both methodologies, have insufficient practical content about developing knowledge, which is suitable for hypothesis testing. We express that deficiency in the left-hand column of Figure 1, with the greyed rectangle containing Develop knowledge for understanding the problem and its solution, and test hypotheses.

We saw in an earlier section, how utility in the business decision-support environment, is determined by executability of results. Despite this, neither methodology contains any final activity, which can be said to produce executable business knowledge, and therefore an executable business solution. We express this deficiency in the greyed rectangle, containing the words *Develop executable business knowledge*. This state of affairs has been described as *the minimisation of interaction between the data miners and the business people* (Pyle 2004, Rule 8).



**Figure 1: Incomplete knowledge development utility of CRISP-DM and DMPM**

We recognise the need for iteration in such projects, but this section of our article is particularly concerned with the absence of key activities, which are required to break the iteration cycle.

Such incompleteness of utility by the project methodology means that:

- the user community must depend on the data mining tool itself to reveal all (Pyle 2004, Rule 4);
- naïve users who depend on the methodologies for producing executable business results, may produce inexecutable results;
- experienced decision-makers, will perceive the incompleteness as a lack of utility, which is required from decision-support project methodologies;
- some business decision-makers, who are unable to distinguish between the technology and its project methodology, will perceive this incompleteness as incapacity of data mining itself, to contribute business value at all.

The lack of knowledge developing business activities in CRISP-DM and DMPM, detract from their decision-support utility, and perpetuates the communication gap between the data mining community and the business community.

## 5 Introducing New Business Domain Knowledge

This is a distinct concept from both methodologies' learning about other data mining solutions for similar problems. It is also distinct from technical myopia, which is

defining the problem in terms of what is familiar in the data (Pyle 2004, Rule 2). Our point in this section is about not defining the business problem *in terms of what is familiar to the business*.

New business domain knowledge represents the new repertoire of understanding and knowledge creation (Pyle 1999, p19). New business knowledge therefore makes the defining and execution of a data mining project possible, which produces breakthrough results.

Neither methodology clearly recognises the need for a simultaneous injection of new business domain knowledge, with the application of data mining technology, for producing executable, breakthrough knowledge. Data mining used in combination with stale, familiar business knowledge, will result in the discovery of information, or the development of knowledge, which is insufficient for meeting the challenges business faces in a changing environment.

The reason is that stale, existing business domain knowledge, has a limiting effect on humans' perception of novelty, causing professional and organisational *paradigm lock*. A paradigm is *the pervasive way of thinking within a group* (McBurney and White 2004, p.20). It follows that paradigm lock is the inability of an organisation to recognise new problems or opportunities, and to solve or realise them. Paradigm lock prevents the formulation and execution of a data mining project with breakthrough potential.

The effects of paradigm lock flow through the knowledge development process:

- limiting the understanding about the business problem or opportunity;
- restricting hypothesis formulation about a number of issues previously identified;
- hampering the identification of information which is required for testing the hypotheses;
- curtailing the identification of the relevant data which should be mined for this information;
- preventing the recognition of hypothetically relevant information even if it were discovered (Lucas 2000, p.29); and
- limiting cognition during knowledge development to what is familiar.

We believe this to be a serious impairment to the decision support utility of both project methodologies.

## 6 Project's Output as Business Value Added

The two methodologies under review, express the output of the project as either discovered knowledge (Chapman, Clinton et al. 1999-2000) or discovered information (SAS Institute 2000). Such terminology does not communicate the results of a data mining project in terms of business value added, and therefore as *beneficial* to the business.

In a previous section, we explained how embedding knowledge developing activities within data mining project methodologies, would progressively add business value within the project. Such an embedding will benefit the methodologies in a further way; it will enable the communication of the outputs of the data mining project, in terms of progressively developed business value, which are the benefits to

the organisation. Communicating the output of data mining projects in terms of their benefits to business, will greatly enhance the uptake of data mining by business.

The consequences of the current state of the methodologies, first is that business decision-makers who already suffer from information overload, may perceive data mining projects as adding to that overload, instead of making sense of information. Second, the perception will continue within resistant sectors of the business community, that the outputs of data mining projects are too complex for development into executable solutions, and therefore added business value. It is our view that these perceptions in business are negatively affecting the uptake of the technology by the business community.

Embedding knowledge developing activities within data mining project methodologies, will greatly improve the utility of the methodologies as communication tools, and will advance the uptake of data mining projects by the business community.

## 7 Detail on Monitor and Control

*Monitor and control* is a very important concept in both the data mining and business literature. Its importance derives from its assurance of the ongoing relevance of solutions, in a changing environment. We have already mentioned that both methodologies have activities for monitoring and controlling. Practically however, in both methodologies little more is accomplished than *statements* that a monitoring plan should be developed. Neither methodology contains substance, about which we can say, that it formulates a monitoring plan.

It is the writer's opinion, that the utility of both methodologies will be enhanced, by drawing on concepts, vocabulary, and practice about monitoring and control, from both the business and the concept drift literature. Both bodies of literature are well-established. The concept drift literature dates back to the mid 80's (Michalski 1987; Widmer and Kubat 1993; Helmhold and Long 1994; Agrawal and Psaila 1995; Harries and Horn 1996; Lanquillon 1999; Klinkenberg and Joachims 2000; Klapper-Rybicka, Schraudolph et al. 2001; Jacobs and Blockeel 2002). A useful data mining methodology should then include an activity, with substance for developing an actual *Monitor and control* plan.

The monitoring section should also include caveats about interpreting the measurements of the monitored parameters, in open problem environments, where not all the influencing factors are captured in the data. The business application of data mining projects mostly falls into this category.

The current state of DMPM and CRISP-DM, assume that a sufficient monitoring plan can - and will indeed - be developed. Considering the complexities of the technology and the knowledge developing activities, and of their interface, the creation of a monitoring plan should not be assumed by any project methodology.

Without an activity for developing a monitoring plan, any data mining project methodology is incomplete. This incompleteness will detract from the utility of such a project methodology. The current incomplete condition of CRISP-DM and DMPM, leaves the business community ill at ease about the data mining community's

appreciation of the importance of monitoring and control in business solution administration. This perception may perpetuate the resistance of the business community to applying data mining technology.

## Conclusion and Future Work

This paper concludes that a number of factors compromise the business decision-support project management utility of DMPM and CRISP-DM. As a result, the business community cannot fully depend on the execution of the results, which these methodologies produce. At the time of their publication, both CRISP-DM and DMPM presented vast improvement over the *status ante*. However, we prognose that as the business community becomes more educated about data mining, the state of the data mining methodologies will increasingly become a limiting factor in the uptake of data mining as a business decision support tool.

The candidate's research aims to address the limitations of the two methodologies, which we evaluated in this article. This research develops a data mining project methodology, which:

- nuances the understanding of the business problem;
- embeds the value adding business knowledge developing activities within a data mining methodology;
- offers a reinterpretation of the sequence of data mining tasks;
- utilises the introduction of new business domain knowledge for developing breakthrough solutions;
- defines a bridging technique between the data mining activities and the business knowledge development activities;
- expresses its output as accrued business value added; and
- adds a *Monitor and control* activity, which uses principles and practise of the concept drift literature, to formulate a monitoring plan.

## Acknowledgements

The researcher is sponsored by an RTS Scholarship. The researcher acknowledges the collaboration of SAS Institute Australia Pty. Ltd., in making DMPM available for this research. He also acknowledges the Australian operation of a multinational mobile phone company, for making available the data, which was used in this research. The researcher acknowledges a sponsorship from the Research Network on Improving Australia's Data Mining and Knowledge Discovery Research to attend AusDM 04.

## References

- 
- Agrawal, R. and G. Psaila (1995). Active Data Mining. First International Conference on Knowledge Discovery and Data Mining, Montreal, Canada, AAAI Press.

- Chapman, P., J. Clinton, et al. (1999-2000). CRISP-DM 1.0: Cross Industry Standard Process for Data Mining, CRISP-DM Consortium.
- Chatfield, C. S. and T. D. Johnson (2000). Step by Step Microsoft Project 2000. Redmond, Washington, Microsoft Press.
- Cooley, R. (2003). Mining Customer Relationship Management (CRM) Data. The Handbook of Data Mining. Y. Nong. London, Lawrence Erlbaum Associates Inc.: 597-616.
- Fayyad, U. (2004). "Editorial." SIKDD Explorations 5(2).
- Ganti, V., J. Gehrke, et al. (1999). "Mining Very Large Databases." IEEE Computer 32(38): 6875.
- Gibson, J. L., J. M. Ivanchevich, et al. (1991). Organisations: Behavior, Structure, Processes. Boston, Irwin.
- Grant, R. M. (1996). "Prospering in Dynamically-Competitive Environments: Organizational Capability as Knowledge Creation." Organization Science 7(4): 375-387.
- Han, J. and M. Kamber (2001). Data Mining: Concepts and Techniques. San Francisco, Morgan Kaufmann Publishers.
- Harries, M. and K. Horn (1996). Learning stable concepts in domains with hidden changes in context. 13th International Conference on Machine Learning, Bari, Italy.
- Hastie, T., R. Tibshirani, et al. (2001). The Elements of Statistical Learning. New York, Heidelberg, Berlin, Springer-Verlag.
- Helmhold, D. P. and P. M. Long (1994). "Tracking Drifting Concepts By Minimizing Disagreements." Machine Learning 14(1994): 27 - 45.
- Jacobs, N. and H. Blockeel (2002). Sequence Prediction with Mixed Order Markov Chains, Department of Computer Science, University of Leuven.
- Klapper-Rybicka, M., N. N. Schraudolph, et al. (2001). Unsupervised Learning in Recurrent Neural Networks. Technical Report, IDSIA (Istituto Dalle Molle di Studi sull'Intelligenza Artificiale). 17.
- Klinkenberg, R. and T. Joachims (2000). Detecting Concept Drift with Support Vector Machines. In Proceedings of the Seventeenth International Conference on Machine Learning, San Francisco, Morgan Kaufmann.
- Kogut, B. and U. Zander (1992). "Knowledge of the Firm, Combinative Capabilities, and the Replication of Technology." Organization Science 3: 383-397.
- Kotler, P. (1988). Marketing Management: Analysis, Planning, Implementation, and Control. Englewood Cliffs, New Jersey, Prentice Hall, Inc.
- Lanquillon, C. (1999). Information Filtering in Changing Domains. International Joint Conference on Artificial Intelligence (IJCAI99), Stockholm, Sweden, August 1999.
- Lucas, J. H. C. (2000). Information Technology for Management. Boston, Irwin McGraw-Hill.
- McBurney, D. H. and T. L. White (2004). Research Methods. Various, Thomson Wadsworth.
- Meltzer, M. (2000). E-Mining Myth and Magic. Using Data Mining Successfully. [www.CRM2day.com](http://www.CRM2day.com), Active Management Techniques.
- Michalski, R. S. (1987). How to learn imprecise concepts: A method employing a two-tiered knowledge representation for learning, Morgan Kauffman.
- Pearce, I. J. A. and J. R. B. Robinson (1991). Strategic Management: Formulation, Implementation, and Control. Boston, Irwin.
- Pyle, D. (1999). Data Preparation for Data Mining. San Francisco, Morgan Kaufmann Publishers.
- Pyle, D. (2004). This Way Failure Lies. DB2 Magazine. 2004.
- SAS Institute (2000). Data Mining Projects Methodology. Cary, NC, SAS Institute Inc.: 133.

- Schön, D. A. (1995). The Reflective Practitioner: How Professionals Think in Action. London, Ashgate Publishing Limited.
- Wedel, M. and W. Kamakura (2000). Segmentation: Conceptual and Methodological Foundations. Boston, Kluwer Academic Publishers.
- Westphal, C. and T. Blaxton (1998). Data Mining Solutions: Methods and Tools for Solving Real-World Problems. New York, Wiley Computer Publishing.
- Widmer, G. and M. Kubat (1993). Effective Learning in Dynamic Environments by Explicit Context Tracking. European Conference on Machine Learning, Springer-Verlag.
- Zikmund, W. G. (2003). Business Research Methods. Various, Thomson South-Western.

## Appendix

Table 1: Pyle's nine rules for failure	
Rule number	Rule
1	Jump right in
2	Frame the problem in terms of the data
3	Focus only on the most obvious ways to frame the problem
4	Rely on your own judgment
5	Find the best algorithms
6	Rely on memory
7	Intuition is more important than standard practice
8	Minimize interaction between miners and business managers
9	Minimize data preparation



# Data Mining Application in a Software Project Management Process

Richi Nayak<sup>1</sup>, and Tian Qiu<sup>2</sup>

<sup>1</sup>School of Information Systems, QUT, Brisbane, QLD 4001, Australia  
r.nayak@qut.edu.au

<sup>2</sup>EDS Credit Services, Adelaide, Australia, tian.qiu@eds.com

*Abstract: During the life cycle of a software project development, many problems are found and raised. Resolutions to these problems are very time consuming and costly. How to use data mining techniques to analyse these problems, and find valuable knowledge to reduce the effort of fixing these problems are discussed in this paper.*

## 1 Introduction

A project leader manages a project with several issues involved such as project planning and scheduling, code implementation, testing and release. It is difficult for him to precisely estimate the project duration in advance. However, he can utilise the Data Mining (DM) methods and make the accurate estimation by learning from the information gained by previous projects. He can also eliminate several potential problems when a pattern appears in his current project similar to the one that have caused problems in previous projects.

Data mining techniques have been successfully applied to various areas such as marketing, medical, and financial [4, 5, 6, 7]. However, few of them can be currently seen in software engineering domain. There exist several difficulties, such as hard to find a data model to put through mining process, no suitable mining tools, poor data quality and acquisition etc. This paper explores the software engineering domain by applying DM techniques to a real world data set. This paper first introduces the undertaken software engineering problem and data mining. After a data model is established for the underlying problem, various data mining techniques are experimented. Interesting findings are discussed together with issues appearing during the mining process.

## 2 Data Acquisition

Every year, more than 50 software projects are carried out in MASC\*, more than ten thousands lines of code are created, and thousands pages of documents are released to various customers. In order to control the problems appearing during a project life

---

\* MASC is a division of a global telecommunication company. A detail of the information source is intentionally removed.

cycle and to improve the working efficiency, MASC has set up a series of processes such as Software Configuration Management, Software Risk Management, Software Project Metric Report and Software Problem Report Management. Data is collected during all these procedures. The Software Problem Report (PR) management data is chosen as the focus of this research. A software bug-tracking system, GNATS (A Tracking System by GNU), is set up on MASC Intranet to collect and maintain all PRs raised from every department and individual within MASC. Currently the GNATS system stores more than 40,000 of problem reports.

Each PR page starts with a number together with a series of information about when, who and which project or department raised this PR. After this, there are several fields that give details of the PR such as *Synopsis*, *Severity*, *Priority*, *Responsible*, *State*, *Class*, *Arrival-Date*, *Closed Date*, *Description*, etc. *Description* details about the bug and its affect the whole system. *Synopsis* is a summary of *Description* field. The *Severity* field shows the criticality of the problem as - *serious*, *critical* or *non-critical*. The *Priority* - *high*, *medium* or *low* – field shows how soon the problem should be resolved. The *State* attribute tells the current stage in the progress of the PR. User can only choose from *open*, *active*, *analysed*, *suspended*, *feedback*, *resolved* and *closed* as the input value of this field. Another important variable is *Class* to state what type the bug is - *sw-bug*, *doc-bug*, *change-request*, *duplicate*, *mistaken*, or *support*.

### 3 Data Pre-processing

A data mining task includes the preprocessing of data before valid, novel, potentially useful, and ultimately understandable patterns are identified in data.

#### 3.1 Defining Goals

MASC engineers make estimations on many aspects of a project such as the number of lines of code to be developed, the kinds of document to be delivered to customer, the time required to accomplish each software engineering stage in the project. There are several tools exist to help a programmer to do the implementation jobs in the software design stage. However, there is few or even no tool exists that can be used for both estimation and project problem reasoning stage. Project team members can only give estimations based on their own experience from previous projects. If the current project is not within their familiar topics, the accuracy of the estimation becomes worse. A PR (problem report) fixing work also becomes more tedious when the responsible person can not estimate the time to fix the problem. Finding a precise estimation figures on bug fixing or estimation work at the early stage of a project will bring great cost savings and accurate progress control to the development team and to the organization.

Currently the GNATS system has no actual database management system implemented. If a PR is closed, it is just statically stored in GNATS and no further analysis is performed. This limits the potential benefits to software engineers who can obtain valuable information if the existing PR data is being analysed. This is useful especially when a programmer is struggling with a bug while a resolution may already hide behind the knowledge that can be derived from the previous similar problems.

These problems can be relieved by utilising data mining techniques. Mining results of the PR data will bring benefits such as accurate project estimation and planning, improved control over the PR fixing and deduction of the project cycle time.

### 3.2 Field Selection

There are several fields such as *Confidential*, *Submitter-ID*, *Environment*, *Fix*, *Release Note*, *Audit Trail*, the associated project name and PR number that are ignored during mining. These fields provide identification information about a PR containing no mining value. Instead, some of these values are used as support roles during pre-processing and post-processing stages to assist in the selection of data and a better understanding of the rules being found.

The aim of this mining exercise is to find useful knowledge from existing projects, all the existing projects should already be finished and all the corresponding PRs should also be closed. Otherwise, a PR can still be changed and is not stable for mining. Hence all PRs with a *closed* value in their *State* field are chosen.

Whenever a PR is raised, a project leader will have to find answers for the following questions before taking any action: How long it will take to fix? How many people were involved? How severe the problem is (customer impact)? What is the impact of the problem on project schedule (Cost & Team priority)? and What type of the problem it is (a Software bug or a design flaw)? Accordingly, attributes such as *Severity*, *Priority*, *Class*, *Arrival-Date*, *Closed-Date*, *Responsible*, and *Synopsis* are considered for mining. The first three attributes describe how a PR is handled within a project, the next three attributes indicate how long a PR is fixed and who were responsible, and the last one lists the content in a PR. The attribute '*class*' is chosen as the target attribute in order to find out any valuable knowledge among the type of a problem and the rest of the PR attributes. Knowing the relationship between the fix effort (in time) and the PR class, a project leader can analyse the fix effort versus the human resources available, and put it in the schedule and resource plan.

The first five fields have fixed input values. *Responsible* attribute is used to calculate how many people were involved to fix the problem. Association or characteristics rules can be found by applying mining techniques on these fields. Whereas *Synopsis* field has no fixed input values. Instead, there is a lot of pure text information stored in this field briefly describing what the problem is in the associated project. It may contain what type of a project document (a piece of code or a support document) that the PR is concerned with. It can be used as a text index. Due to the nature of the values inside this field, a different mining approach, Text Mining is considered to deal with such kind of text values.

### 3.3 Data Cleaning

Data is further investigated to identify problems, such as missing values, inconsistent values, and mistaken values using graphical tools such as histogram for frequency distribution of the values, calculating maxima, minima and mean values. Histogram plots the contribution made by each value for the (categorical) attribute, and therefore helps to identify distribution skews and invalid values. The occurrence of these problems comes with several factors such as human mistakes and evolutions of the

GNATS system. An example is the use of different terminologies over the time such as *SW-bug* or *sw-bug* as an input value for *Class* field (Example a, d in Figure 1). A *Time-Zone* field and other new input values have been added later in the system on management request based on feedback of users after several years of system running.

For the PRs in which an error can be recovered manually or automatically by software, the modified PRs are included in the mining process. For example, *SW-bug* in *Class* field is replaced by *sw-bug* throughout the data. Another example is filling the data in wrong fields such as mistakenly input the closing date in the obsolete *Completed-Date* field instead of the *Closed-Date* field.

The PRs, in which an error cannot be recovered precisely, are either discarded or replaced by a '?' if a software can handle the missing values. For example, the pattern a in Figure 1 has its closed time earlier than the time being raised. Some PRs do not have all the values stored, such as Example c in Figure 1 has no closed date. An example of inconsistent values is shown in Figure 1 - there is no input for the *Time-Zone* field in a PR recorded before 1998, as the *Time-Zone* field is added in 1998.

<i>PR_ID</i>	<i>Category</i>	<i>Severity</i>	<i>Priority</i>	<i>Class</i>	<i>Arrival-Date</i>	<i>Close-Date</i>	<i>Synopsis</i>	
a.	17358	lbamb	serious	high	sw-bug	20:50 May 25 CST 1999	11:35 Mar 24 CST 1999	STI STR register not being reset at POR
b.	17436	lbamb	serious	high	support	18:10 Mar 30 CST 1999	12:00 May 24 CST 1999	sequence_reg variable in the RDR_CHL task is not defined
c.	580	lbngarr	serious	low	doc-bug	10:10 May 31 May 1996		In URDRT2 of design doc, the word 'last' should be 'first'
d.	6205	lgali	serious	medium	SW-bug	14:30 Nov 5 1997	13:14 Dec 1 1997	grouping of options in dialog box

Figure 1: Data examples from the original PR data set

<i>Severity</i>	<i>Priority</i>	<i>Time-to-fix</i>	<i>Class</i>	<i>Synopsis</i>
a.	serious, high, 61,	sw-bug,	STI STR register not being reset at POR	
b.	serious, high, 56,	support,	sequence_reg variable in the RDR_CHL task is not defined	
c.	serious, low, ?,	doc-bug,	In URDRT2 of design doc, the word 'last' should be 'first'	
d.	serious, medium, 24,	sw-bug,	grouping of options in dialog box	

Figure 2: Data examples ready for mining

### 3.4 Data Transformation

Data transformation is considered, in the way of converting attributes *Arrival-Date* and *Closed-Date* to a time-period - identifying the time spent to fix a PR - taking account the additional information *Time-Zone* and *Responsible*. This transformation resulted in the *Time-to-fix* attribute with continuous values (figure 2). The *Responsible* attribute

has the information about personnel engaged in rectifying the problem. We assume that the derived attribute *Time-to-fix* is total time spent to fix a problem if there is only one person involved. The calculated time period from *Arrival-Date* and *Closed-Date* is then multiplied by the number of people yield from the *Responsible* attribute. In order to improve the mining quality, we have discretized this attribute with cutting points be one day (1), half week (3 days), one week (7), two weeks (14), one month (30) and one quarter (90 days), half year (180 days) and more than one year (360 days). So that the mining results are not very highly depended on the exact human resource involved but gives an approximate estimate, allowing a minor change in human resource.

#### 4 Data Modelling and Mining

Out of total 40000 PRs initially selected as the data set, we are left with 11,000 PRs after pre-processing (figure 2). These 11,000 PRs (depends on different mining tasks, the numbers varies a little) cover more than 120 projects within MASC from 1996 to 2000. For example 11364 PR records have been applied with text-mining tools on the valid values in their Synopsis fields, as 364 records have no time values so could not be used for classification task.

The GNATS system provides simple methods to retrieve basic information from the PR data set, such as the PR numbers related to a particular person, etc. Besides this, general database query languages (such as SQL) can also give useful information, i.e., the average time spent for fixing a PR in a project. However these methods cannot perform if user likes to (1) pose queries on a large number of records with high dimensional structures, (2) summarise a large data set to facilitate decision-making, (3) make predictions on new data based on the existing rules, and (4) visualise simplified extracted local structures. On the contrary, data mining techniques perform well on these cases, and are able to reveal the deeper characters of the data, such as:

- If a PR is raised, how long should it take to fix the problem?
- What type of project documents needs a significant effort to fix an associated bug?

The selection of data mining operations and techniques is one of the most important things that directly affect the progress and the accomplishment of any DM applications. We have chosen:

- Prediction modelling on the time consuming patterns of the PR data to make estimation.
- Link analysis to discover association among the contents/values of the variables being selected.
- Text Mining to *analyse Synopsis* field.

The predictive modelling or classification task builds a model on existing dataset by recognising distinct characteristics of the data set. The built model predicts future events based on previous data, specifying a class (or label) to each record in the dataset. A supervised machine learning algorithm, that learns a model on previous or existing data, can be used for predictive modelling task. These models are developed over training and testing phases. The model is given some already known facts with correct answers during the training phase, from which the model learns to make accurate predictions. During the testing phase, the model is exposed to new data set to check the predictive capability. Various classification methods are neural induction,

tree induction and bayesian classifiers, K-nearest neighbour classifiers, case based reasoning, genetic algorithms, rough set and fuzzy set approaches [4, 5, 6, 7].

Tree induction or decision tree is used for this task. Decision tree has been quite popular in data mining due to their simplicity, efficiency and capability of dealing with a large number of training examples. The decision tree learning algorithms start by constructing a decision tree from top to bottom. Attributes are evaluated at each step to form descendant nodes. The attribute selection is based on a 'statistical test' to determine how well it classifies the training examples. An internal node represents a test on an attribute and a leaf node represents a class or class distribution. Classification of unknown samples is made by tracing a path through the decision tree until a leaf node having the class prediction is reached. The maximum height of the decision tree depends on the number of attributes used to define the rules. Because the number of attributes in our problem is small, the resulting decision tree is relatively simple and thus its structure is understood easily by a human analyst.

The link analysis operation exposes samples and trends by predicting correlation of variables in a given data set. Association discovery builds a model to find variables implying the presence of other variables (with a certain degree of confidence and support) in the given data set. This process reveals hidden affinity among the variables i.e. which variables cause one another if a PR report is being raised. The technique is based on counting occurrences of all possible combination of variables. Apriori and its variation algorithms [6] are most widely used.

In general, Data mining is knowledge discovery from structured databases. Text mining techniques, on the other hand, discover the knowledge from an unstructured textual data. In order to discover and use the implicit structure of the texts, text mining techniques integrate some specific natural language processing to pre-process the textual data. A suitable data-mining tool should satisfy the ease of use, low cost, ease of preparation and appropriate for the data model [5]. The C5 [2] is used for classification, CBA [1] is used for both classification and association rules mining, and TextAnalyst [3] is used for text mining.

## 5 Assimilation and Analysis of Outputs

### 5.1 Classification and Association Rule Mining

In order to get better rules and to decrease the error rate as much as possible, several approaches are used. One approach is to stratify the data on the target using the choice-based sampling rather than using random samples. Equal number of samples representing each possible value of the target attribute (*Class*) is chosen for training. This improves the possibility of finding rules that are associated with small group of values during training. Another approach is to choose different number of PR data as training sets. We use three different training data sets. The first data set (Case 1, Table 1) chooses 1224 PRs from only one software project. The second one (Case 2) builds equal distributed value for a medium size of 3400 PRs (1000 PRs from each value of '*Class*') from all software projects. The third data set (Case 3) contains a large size of 5381 PRs from all software projects.

We use two learning engines to discover rules from the PR data set– single support CBA (labelled a in the Table 1 e.g., Case1a) and multiple support CBA (labelled b in

the Table 1 e.g., Case1b). Constraints, support and confidence, are included in rules to control the quality of results. Confidence is the measure of the strength of a rule that indicates the probability of having consequence(s) in the rules provided that the rule contains certain antecedent(s). Support indicates the number of input data supporting the rule. Since, users are interested in rules with worth consideration or more preferred or more certain, a threshold for support and confidence is set.

Setting a threshold for minimum support and confidence is a result of trial and error. If these factors are set too high, very few rules may be discovered. If the factors are set too low, too many rules may be generated with very low values. Under certain situation, attributes in the data are not likely to have uniform distributions, and many attributes are of very low frequency. Therefore a single support for all attributes may not be able to discover important rules that involve such rare attributes (very low frequency). This problem is relieved by setting multiple supports that allow user to choose different minimum supports to different attributes. Table 1 reports the classification mining results on all three cases and the associative rule mining results as Case4. The test data is 10% of the whole data set and chosen randomly with the special consideration that each output class is representing in the test data.

Table 1: CBA Mining Results Summary. Rules are ranked by confidence.

	#Rules	Error rate (%)		Time cost (seconds)	
		Training	Testing	Training	Testing
Case1a	10	45.180	47.56	1.01	0.07
Case1b	9	45.180	47.56	1.04	0.09
Case2a	41	57.04	51.95	0.41	1.1
Case2b	21	59.10	58.25	0.44	1.0
Case3a	20	43.51	43.5	2.2	2.0
Case3b	15	46.5	45.1	1.6	1.9
Case4a	10	45.180	N/A	0.66	N/A
Case4b	9	45.180	N/A	1.04	N/A

In general, all classification data mining operations in CBA software achieve around 46% Error rate in training data set (the lowest is 43.51%, the highest is above 59.10%). Above 51% correct prediction rate is achieved in testing data set (the lowest has 43.51%, the highest has 58.25%). Another interesting point is that the attempt to improve the accurate prediction in the way of equal-distributed target-value samples does not lead much change; there is only roughly 3% improvement over the final result. The error rates from using multiple supports are higher and the number of extracted rules is lower than those from using single support mining engine. There is no rule that has confidence value larger than 80%, however they do describe some characters of the PR fixing patterns. Therefore they are useful for software project management in estimation bug fixing related time issues.

Followings are examples of generated classification rules with CBA:

Rule 1: If *severity*= *non-critical* and *Time-to-fix* = 3 to 30 days and *priority*= *medium*  
Then *class* = *doc-bug*. Confidence = 82.7%, Support = 2.7%

Rule 2: If *severity = critical* and *Time-to-fix = less than 3 days* and *priority = high*  
Then *class = sw-bug*. Confidence = 75.2%, Support = 2.3%

Overall the extracted rules conclude that software related bugs can be fixed within 3 days with above 75% confidence if they have high priority and are in critical condition. It may take 3 months to fix the problem if the corresponding priority and severity are graded as medium and serious. Certainly, the confidences of the rules are low, and only a small number of cases support these rules.

Table 2: C5 Mining Results Summary

	Normal mining		Mining with Boosting		Mining with cross-validation (10-fold)	
	Training	Testing	Training	Testing	Training	Testing
#Rules	51	11	N/A.	N/A	57.7	12.4
Error Rate (%) (Rules)	41.5	42.6	41.3	42.6	43.9	42.8
Error Rate (%) (Trees)	40.3	42.5	39.4	42.6	44.1	43.1
Size of tree	141	21	N/A.	N/A	121.9	17.4
Process Time (seconds)	5.6	0.2	37.7	0.4	41.1	1.1

The software C5 was also used to perform classification data mining with the boosting and cross validation techniques (Table 2). The cross validation technique splits the whole data set into several subsets (called folds). Let each fold to be the test case and the rest as training sets in turn during training. Boosting is a technique for generating and combining multiple classifiers to give improved predictive accuracy. After a number of trials, several different decision trees or rule sets are combined to reduce error rate for prediction. Boosting takes a longer time to produce the final classifier, and may not always achieve better results than a single classifier approach does, especially when the training data set has noise. Boosting and cross validation techniques do not generate a new rule, but try to find a better rule from the existing results. They only produce better results than the individual trees if the individual trees disagree with one another. Some example extracted classification rules with C5 are:

Rule 1: When a PR is in *low priority* and the *time spent is around half a day (0.5 day)*

Then the rule has a high probability (87.5% Confidence) to classify a bug to be a *document related bug*.

Rule 2: When a PR is in *medium priority* with *non-critical severity* and the *time spent is around 1.1 day* Then the rule has 84.6% Confidence to classify a bug to be a *document related bug*.

Rule 3: When a PR is in *low priority* and the *time spent for fixing is around 1 week*

Then the rule has 83.3% Confidence to classify a bug to be a *software bug*.

In general, all data mining operations achieves around 42% error rate in rules from the training set (the lowest is 40.3%, the highest is 43.9%). Similar error rate value is

achieved for the generated trees in testing data set (the lowest is 39.4%, the highest is 44.1%). Both of the rates are better than CBA results. The time efficiency of C5 is also better than CBA.

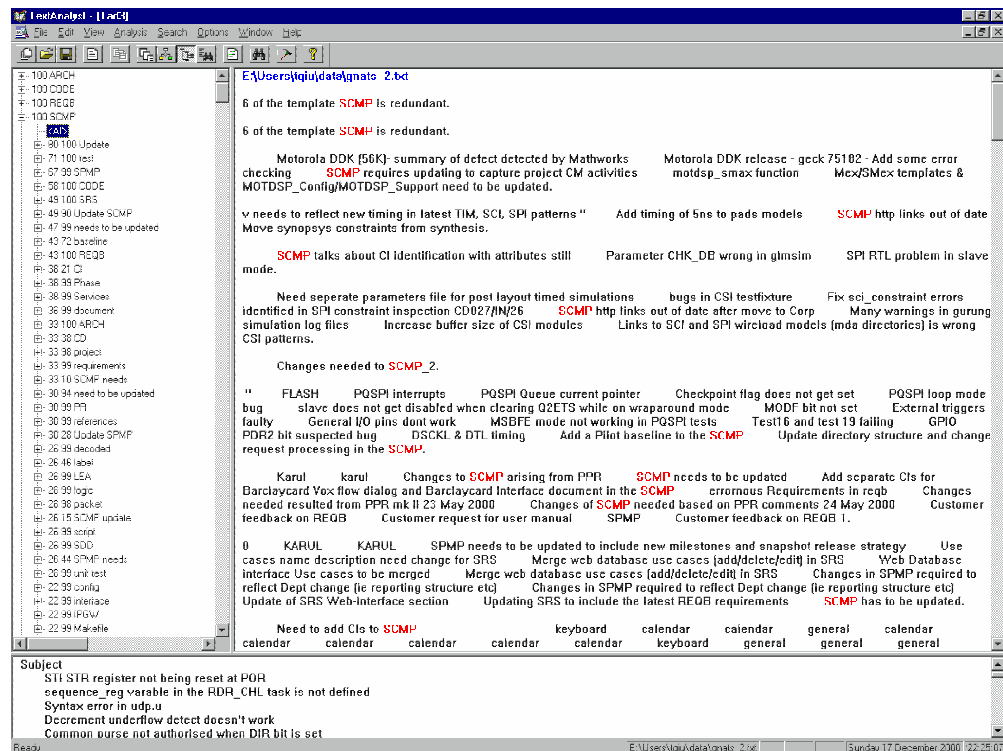


Figure 3: Text Analyst Mining: An interface

## 5.2 Text Mining in PR data

In order to find valuable knowledge from thousands lines of text, we categorise the pure text into several document types based on certain background knowledge. The analysis of the text together with the rules obtained from classification and association can more accurately predict the time and cost of fixing PRs. We used TextAnalyst [3] tool to automatically summarise the pure text data and extract some valuable rules. It builds up a semantic network for the investigation over the PR data. Each element of the semantic network is characterised by a weight value and a set of relationship of this element to other elements in the network. Every relationship between elements is also assigned a weight value. The semantic network can then provide a concise and accurate summary of the analysed text.

TextAnalyst automatically creates the semantic network based on the structure, vocabulary and volume of the analysed text, without any predefined rules. The semantic network tree of the PR data (figure 3) contains a set of the most important words or word combinations, called *concepts*. The relations among those concepts

together with the semantic weights of concepts and relations are also shown. The values of the weights range from 0 to 100, which correspond to the probability that the associated concept is characteristic for the whole PR data. It also shows all the text related to a concept if the user clicks the corresponding concept.

Text mining is applied on a total 11226 cases. An interesting result is obtained for SMTP, Software Configuration Management Plan, a support document in every MASC project. Since SCMP is not a main design document for a project with just about tens pages, it has never been considered as a trouble making item. But amazingly, the result showed that SCMP has 71 percentage probability of appearing in test related PR records, and 58 percentage probability of appearing in Code related PR records. This result is even higher than the result associated with SRS (“Software Requirements Specification”, a main development document directly related to software). Although we can not say SCMP causes more problems than SRS, but the higher appearance of SCMP inside test related PR definitely shows a warning. It is worth for software engineers to be more careful when dealing with SCMP document, and hence reducing the total cost of fixing SCMP related PRs. Another analysis shows that a test related PR has even a higher weight (36, 100) in document related bugs than a SRS related PR (35, 99) does. Which suggests a better project management should not only focus on the quality of product related documents, but also pay attention on the quality of testing related documents.

### 5.3 Existing problems in performing mining

The error rates of testing data sets in both CBA and C5 are higher than expected. Although several approaches are attempted to reduce the error such as uniform distribution of values, cross validation, boosting, different size of training set, etc. Unfortunately, the average error rate is only fallen down by 5% from 47% to 42%. The best result is 9% decrease from 46% to 37%. These results indicate that some amount of noise is still existent in data after dealing with the noise during pre-processing.

For example, the relationship between PRs and human resources within a particular project plays a great impact. The time needed to fix a bug is different for each project depending upon the actual human resources available. We have used only the attribute ‘*Responsible*’ to indicate the human resource available. Truly, the relationship with the human resources available for past projects whose data was analysed is needed to use time patterns to help project leaders to predict time consumptions more accurately. The use of additional data source ‘Change Request data set’ that records all customer request process data may rectify this problem.

Another reason is a non-uniform value distribution. For example, there are only 342 PRs with *change-request* value in the data set, compared to more than 5900 PRs related to *sw-bug*. Any potential rule associated to *change-request* can be heavily affected due to the presence of large size group with other values. Again the use of additional data sources together with the PR data set can rectify this problem. We also attempted to use neural network techniques, but it is difficult to interpret the outputs.

## 6 Future Directions and Conclusion

This paper explored the use of data mining techniques on a set of data collected from the software engineering process under a real software business environment. Some useful rules are inferred on the time patterns of the PR fixing and the relationship between the content and the type of a PR in the form of association rules, classification rules or semantic trees.

The time patterns rules may help a project leader to estimate or predict time consumptions more accurately than before. Another finding suggests that bug fixing efforts have more probabilities to be spent on test related PR. This could cost the project team a lot of time in fixing non-product-related problems. By giving more attention in design and development of these documents, the project efficiency will be improved.

Results of the application indicate that data mining techniques bring more power to improve the quality and efficiency of the software development process, even though the scale of the data mining task is limited. It will be interesting to apply data mining to different phases of software development such as software quality data, etc. As in many other domains, the benefits and capabilities brought by data mining in software engineering domain are worth of further investigations.

## 7 References

1. CBA, <http://www.comp.nus.edu.sg/~dm2/>
2. C5.0, <http://www.rulequest.com/see5-info.html>
3. TextMiner, <http://www.megaputer.com/company/index.html>
4. P.Cabena, P. Hadjinian, R. Stadler, J. Verhees, and A. Zanasi, “*Discovering Data Mining: From Concept to Implementation*”, Prentice Hall, 1997.
5. H. Edelstein, “*Mining for gold. (Selecting data mining tools)*”, Information Week, April 21, 1997 n627 p53 (6).
6. J. Han and M. Kamber, “*Data mining: concepts and techniques*”, Morgan Kaufmann Publishers, 2001.
7. C. Westphal, and T. Blaxton, “*Data Mining Solutions: Methods and Tools for Solving Real-World Problems*”, John Wiley & Sons, Inc, 1998.



# A Probabilistic Geocoding System based on a National Address File

Peter Christen<sup>1\*</sup>, Tim Churches<sup>2</sup>, and Alan Willmore<sup>2</sup>

<sup>1</sup> Department of Computer Science, Australian National University,  
Canberra ACT 0200, Australia, [peter.christen@anu.edu.au](mailto:peter.christen@anu.edu.au)

<sup>2</sup> Centre for Epidemiology and Research, New South Wales Department of Health,  
Locked Mail Bag 961, North Sydney NSW 2059, Australia,  
[{tchur,awill}@doh.health.nsw.gov.au](mailto:{tchur,awill}@doh.health.nsw.gov.au)

**Abstract.** It is estimated that between 80% and 90% of governmental and business data collections contain address information. Geocoding – the process of assigning geographic coordinates to addresses – is becoming increasingly important in many application areas that involve the analysis and mining of such data. In many cases, address records are captured and/or stored in a free-form or inconsistent manner. This fact complicates the task of robustly matching such addresses to spatially-annotated reference data. In this paper we describe a geocoding system that is based on a comprehensive high-quality geocoded national address database. It uses a learning address parser based on hidden Markov models to separate free-form addresses into components, and a rule-based matching engine to determine the best set of candidate matches to a reference file. The geocoding software modules are implemented (as part of the *Febri* open source data linkage system) in the object-oriented language Python, which allows rapid prototype development and testing.

**Keywords:** Data mining preprocessing, geocoding, spatial data analysis, data linkage, data cleaning, indexing, G-NAF, hidden Markov model.

## 1 Geocoding

Increasingly, many data mining and data analysis projects need information from multiple data sources to be integrated, matched, combined or linked in order to enrich the available data and to allow more detailed analysis. The aim of such linkages is to merge all records relating to the same entity, such as a patient, customer or business. Most of the time the linkage (or matching) process is challenged by the lack of a common unique entity identifier, and thus becomes non-trivial [3, 8, 15]. In such cases, the available partially identifying information – like names, addresses, and dates of birth – is used to decide if two (or more) records correspond to the same entity. This process is compute intensive, and linking today's large data collections becomes increasingly difficult using traditional linkage techniques.

---

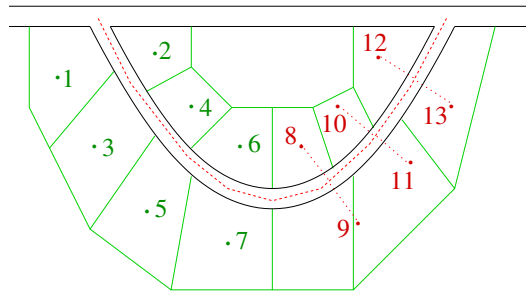
\* Corresponding author

A special case of linkage is *geocoding*, the matching of a data source with geocoded reference data (which is made of cleaned and standardised records containing address information plus their geographical location). The US Federal Geographic Data Committee estimates that geographic location is a key feature in 80% to 90% of governmental data collections [14]. In many cases, addresses are the key to spatially enable data. The aim of geocoding is to generate a geographical location (longitude and latitude) from street address information in the data source. Once geocoded, the data can be used for further processing, in spatial data mining [6] projects, and it can be visualised and combined with other data using Geographical Information Systems (GIS).

The applications of spatial data analysis and mining are widespread. In the health sector, for example, geocoded data can be used to find local clusters of disease. Environmental health studies often rely on GIS and geocoding software to map areas of potential exposure and to locate where people live in relation to these areas. Geocoded data can also help in the planning of new health resources, e.g. additional health care providers can be allocated close to where there is an increased need for services. An overview of geographical health issues is given in [1]. When combined with a street navigation system, accurate geocoded data can assist emergency services find the location of a reported emergency (for example, if a caller reports an incomplete or unclear address).

Geocoded customer data, combined with additional demographic data, can help businesses to better plan marketing and future expansion, and the analysis of historical geocoded data, for example, can show changes in their customer base. Within census, geocoding can be used to assign people or households to small area units, for example census collection districts, which are then the basis of further statistical analysis.

There are two basic scenarios for geocoding user data. In the first, a user wants to automatically geocode a data set. The geocoding system should find the *best possible* match for each record in the user data set without human intervention. Each record needs to be attributed with the corresponding location plus a *match status* which indicates the accuracy of the match obtained (for example an exact address match, or a street level match, or a postcode level match). This scenario might become problematic if the user data is not of high quality, and contains records with missing, incorrect or out-of-date address information. Typographical errors are common with addresses, especially when they are recorded over the telephone or from hand-written forms. As reported in [11], a match rate of 70% successfully geocoded records is often considered an acceptable result. In the second scenario a user wants to geocode a single address that may be incomplete, erroneous or unformatted. The system should return the location if an exact match can be found, or alternatively a list of possible matches, together with a matching status and a likelihood rating. This geocoding of a single record should be done in (near) real time (i.e. less than a couple of seconds response time) and be available via a suitable user interface (e.g. a Web site).

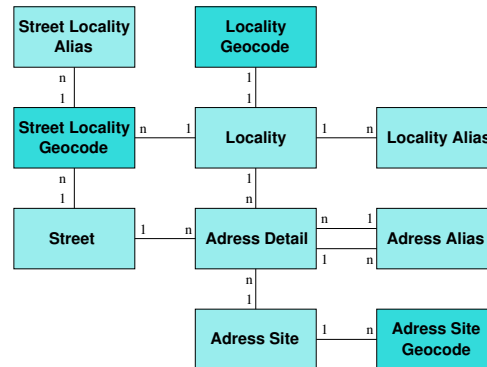


**Fig. 1.** Example geocoding using property parcel centres (numbers 1 to 7) and street reference file centreline (dashed line and numbers 8 to 13, with the dotted lines corresponding to a global street offset).

Standard data (or record) linkage techniques [3, 8, 15], where the aim is to link (or match) together all records belonging to the same entity, normally classify compared record pairs into one of the three classes *links*, *non-links* and *possible links*, with the latter class containing those record pairs for which human oversight, also known as *clerical review*, is needed to decide their final linkage status. Often no additional information is available so the clerical review process becomes one of applying human intuition, experience or common sense to the decision based on available data. This is similar to the second geocoding scenario described above, where the user is presented with a selection of possible matches (sorted according to their matching status and likelihood rating).

Many GIS software packages provide for street level geocoding. As a recent study shows [2], substantial differences in positional error exist between addresses which are geocoded using street reference files (containing geographic centreline coordinates, street numbers and names, and postcodes) and the corresponding true locations. The use of point property parcel coordinates (i.e. the centres or centroids of properties), derived from cadastral data, is expected to significantly reduce these positional errors. Figure 1 gives an illustrative example. Even small discrepancies in geocoding can result in addresses being assigned to, for example, different census collection districts, which can have huge implications when doing small area analysis. A comprehensive property based database is now available for Australia: the Geocoded National Address File (G-NAF). It is presented in details in Section 1.1.

We give an overview of our geocoding system in Section 2. The two central technical issues for a geocoding system are (1) the accurate and efficient matching of user input addresses with the address information stored in the geocoded reference data, and (2) the efficient retrieval of the address location (longitude and latitude) of the matched geocoded records. In order to achieve accurate match results, addresses both in the user data set and the geocoded reference data need to be cleaned and standardised in the same way. We cover this issue in more details in Section 2.1. Address locations can efficiently be retrieved from



**Fig. 2.** Simplified G-NAF data model (10 main files only). Links  $1-n$  denote one-to-many, and links  $1-1$  denote one-to-one relationships.

the geocoded reference data by converting the traditional database tables (or files) into inverted indexes, as presented in Section 2.2. The geocode matching engine is the topic of Section 3, with some initial experimental results presented in Section 4, and conclusions and an outlook to future work is given in Section 5.

### 1.1 G-NAF – A Geocoded National Address File

In many countries geographical data is collected by various state and territory agencies. In Australia, for example, each state and territory have their own governmental agency that collect data to be used for land planning, as well as property, infrastructure or resource management. Additionally, national organisations like post and telecommunications, electoral rolls and statistics bureaus collect their own data. All these data sets are collected for specific purposes, have varying content and are stored in different formats.

The need for a nation-wide, standardised and high-quality geocoded data set has been recognised in Australia since 1990 [11], and after years of planning, collaborations and development the G-NAF was first released in March 2004. Approximately 32 million address records from 13 organisations were used in a five-phase cleaning and integration process, resulting in a database consisting of 22 normalised files (or tables). Figure 2 shows the simplified data model of the 10 main G-NAF files.

G-NAF is based on a hierarchical model, which stores information about address sites separately from locations and streets. It is possible to have multiple geocoded locations for a single address, and vice versa, and aliases are available at various levels. Three geocode files contain location (longitude and latitude) information for different levels. If an exact address match can be found, its location can be retrieved from the ADDRESS\_SITE\_GEOCODE file. If there is only a match on street level (but not street number), the STREET\_LOCALITY\_GEOCODE file will

**Table 1.** Characteristics of the 10 main G-NAF files (NSW data only).

G-NAF data file	Numbers of records and attributes	Keys (persistent identifiers)
ADDRESS_ALIAS	289,788 / 6	PRINCIPAL_PID
		ALIAS_PID
ADDRESS_DETAIL	4,145,365 / 28	GNAF_PID
		LOCALITY_PID
		STREET_PID
		ADDRESS_SITE_PID
ADDRESS_SITE	4,096,507 / 6	ADDRESS_SITE_PID
ADDRESS_SITE_GEOCODE	3,336,778 / 12	ADDRESS_SITE_PID
LOCALITY	5,017 / 7	LOCALITY_PID
LOCALITY_ALIAS	700 / 5	LOCALITY_PID
		ALIAS_PID
LOCALITY_GEOCODE	4,978 / 11	LOCALITY_PID
STREET	58,083 / 6	STREET_PID
STREET_LOCALITY_ALIAS	5,584 / 6	STREET_PID
		LOCALITY_PID
STREET_LOCALITY_GEOCODE	128,609 / 13	STREET_PID
		LOCALITY_PID

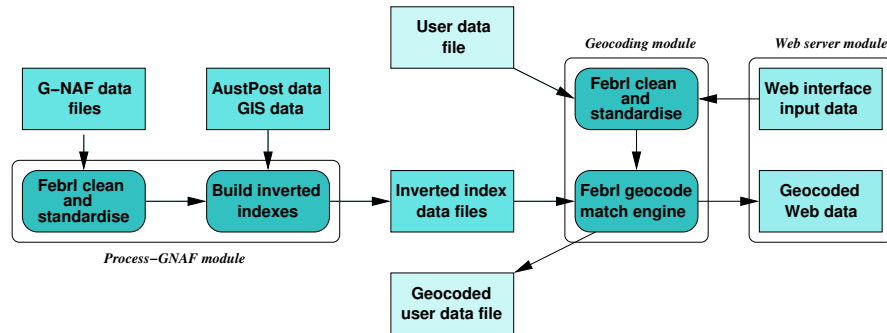
provide an overall street geocode. Finally, if no street level match can be found the `LOCALITY_GEOCODE` file contains geocode information for localities (e.g. towns and suburbs). Both the `STREET_LOCALITY_GEOCODE` and `LOCALITY_GEOCODE` files also contain information about the extent of streets and localities.

For our project we only used the G-NAF records covering the Australian state of New South Wales (NSW), containing around 4 million address, 60,000 street and 5,000 locality records. Table 1 gives an overview of the size and content of the 10 main G-NAF data files used.

## 2 System Overview

The geocoding system presented in this paper is part of the *Febrl* (Freely Extensible Biomedical Record Linkage) data linkage system [3, 7], that contains modules to clean and standardise data sets which can contain names, addresses and dates; and link and deduplicate such cleaned data. An overview of the *Febrl* geocoding system is shown in Figure 3. The geocoding process can be split into the preprocessing of the G-NAF data files (which is described in detail in Sections 2.1 and 2.2), and the matching with user-supplied addresses as presented in Section 3.

The preprocessing step takes the G-NAF data files and uses the *Febrl* address cleaning and standardisation routines to convert the detailed address values (like street names, types and suffixes, house numbers and suffixes, flat types



**Fig. 3.** Overview of the *Febrl* geocoding system.

and numbers, locality names, postcodes, etc.) into a form which makes them consistent with the user data after *Febrl* standardisation. Note that the G-NAF data files already come in a highly standardised form, but the finer details, for example how whitespaces within locality names are treated, make the difference between successful or failed matching. The cleaned and standardised reference records are then inserted into a number of inverted index data structures.

Additional data used in the preprocessing step are a postcode-suburb look-up table which is publicly available, and which can be used to impute missing postcodes or suburb values in the G-NAF locality files; and a table extracted from a commercial GIS system containing postcode and suburb boundary information, which is used to create *neighbouring region* look-up tables.

The geocode matching engine takes as input the inverted indexes and the raw user data, which is cleaned and standardised before geocoding is attempted. As shown in Figure 3, the user data can either be loaded from a data file, geocoded and then stored back into a data file, or it can be passed as one or more address(es) to the geocoding system and returned via a Web interface.

The complete *Febrl* system, including the geocoding and Web server modules, is implemented in the object-oriented open source language *Python*<sup>1</sup>, which allows rapid prototype development and testing.

## 2.1 Probabilistic Address Cleaning and Standardisation

The first crucial step when processing both the geocoded reference files and the user data is the cleaning and standardisation of the data (i.e. addresses) used for geocoding. It is commonly accepted that real world data collections contain erroneous, incomplete and incorrectly formatted information. Data cleaning and standardisation are important preprocessing steps for successful data linkage and before including such data in a data warehouse for further analysis [13]. Data may be recorded or captured in various, possibly obsolete, formats and data items may

<sup>1</sup> <http://www.python.org>

be missing, out-of-date, or contain errors. The cleaning and standardisation of addresses is especially important for data linkage and geocoding so that accurate matching results can be achieved.

The main task of cleaning and standardising addresses is the conversion of the raw input data into well defined, consistent forms and the resolution of inconsistencies in the way address values are represented or encoded. Rule-based data cleaning and standardisation is currently used by many commercial systems and is cumbersome to set up and maintain, and often needs adjustments for new data sets. We have recently developed (and implemented within *Febrl*) new probabilistic techniques [4] based on hidden Markov models (HMMs) [12] which achieved better address standardisation accuracy and are easier to set-up and maintain compared to popular commercial linkage software.

A HMM is a probabilistic finite-state machine consisting of a set of observation or output symbols, a finite set of discrete, hidden (unobserved) states, a matrix of transition probabilities between those hidden states, and a matrix of probabilities with which each hidden state emits an observation symbol [12] (this *emission matrix* is also called an *observation matrix*). In our case, the hidden states of the HMM correspond to the output fields of the standardised addresses.

The *Febrl* approach to address cleaning and standardisation consist of the following three steps.

1. The user input addresses are *cleaned*. This involves converting all letters to lower-case, removing certain characters (like punctuations), and converting various sub-strings into their canonical form, for example 'c/-', 'c/o' and 'c.of' would all be replaced with 'care\_of'. These replacements are based on user-specified and domain specific substitution tables. Note that these substitution tables can also contain common misspellings for street and locality names, for example, and thus help to increase the matching quality.
2. The cleaned input strings are split into a list of words, numbers and characters, using whitespace marks as delimiters. Look-up tables and some hard-coded rules are then used to assign one or more tags to the elements in this list. These tags will be the observation symbols in the HMM used in the next step.
3. The list of tags is given to a HMM, and assuming that each tag (observation symbol) has been emitted by one of the hidden states, the *Viterbi* algorithm [12] will find the most likely path through the HMM, and the corresponding hidden states will give the assignment of the elements from the input list to the output fields.

Consider for example the address '73 Miller St, NORTH SYDNEY 2060', which will be cleaned (SYDNEY corrected to *sydney*), split into a list of words and numbers, and tagged in steps one and two. The resulting lists of words/numbers and tags looks as follows.

```
['73', 'miller', 'street', 'north_sydney', '2060']
['NU', 'UN', 'WT', 'LN', 'PC']
```

with 'NU' being the tag for numbers, 'UN' the tag for unknown words (not found in any look-up table or covered by any rule), 'WT' the tag for a word found in

the wayfare (street) type look-up table, 'LN' the tag for a sequence of words found to be a locality name, and 'PC' the tag for a known postcode.

In the third step the tag list is given to a HMM (which has previously been trained using similar address training data), and the *Viterbi* algorithm will return the most likely path through the HMM which will correspond to the following sequence of output fields.

```
'street number': '73'
'street name': 'miller'
'street type': 'street'
'locality name': 'north_sydney'
'postcode': '2060'
```

Details about how to efficiently train the HMMs for address (as well as name) standardisation, and experiments with real-world data are given in [4]. Training of the HMMs is quick and does not require any specialised skills. For addresses, our HMM approach produced equal or better standardisation accuracies than a widely-used rule-based system.

## 2.2 Processing the G-NAF Files

Processing the G-NAF data files consists of two steps, the first being the cleaning and standardisation as described above, and the second being the building of inverted indexes. Such an inverted index is a keyed hash-table in which the keys are the values from the cleaned G-NAF data files, and the entries in the hash-table are sets with the corresponding PIDs (persistent identifiers) of the values. For example, assume there are four records in the `LOCALITY` file with the following content (the first line is a header-line with the attribute names).

locality_pid,	locality_name,	state_abbrev,	postcode
60310919,	sydney,	nsw,	2000
60709845,	north_sydney,	nsw,	2059
60309156,	north_sydney,	nsw,	2060
61560124,	the_rocks,	nsw,	2000

The inverted indexes for the three attributes `locality_name`, `state_abbrev` and `postcode` then are (square brackets denote lists and round brackets denote sets):

```
locality_name_index = ['north_sydney':(60709845,60309156),
                      'sydney':(60310919),
                      'the_rocks':(61560124)]

state_abbrev_index = ['nsw':(60310919,60709845,60309156,61560124)]

postcode_index = ['2000':(60310919,61560124),
                  '2059':(60709845),
                  '2060':(60309156)]
```

The matching engine then finds intersections of the inverted index sets for the values in a given record. For example, a postcode value '2000' would result in a set of PIDs (60310919,61560124), and when intersected with the PIDs for

**Table 2.** G-NAF attributes used for geocode matching.

G-NAF data file	Attributes used
ADDRESS_DETAIL	flat_number_prefix, flat_number, flat_number_suffix, flat_type, level_number, level_type, building_name, location_description, number_first_prefix, number_first, number_first_suffix, number_last_prefix, number_last, number_last_suffix, lot_number_prefix, lot_number, lot_number_suffix
LOCALITY_ALIAS	locality_name, postcode, state_abbrev
LOCALITY	locality_name, postcode, state_abbrev
STREET	street_name, street_type, street_suffix
STREET_LOCALITY_ALIAS	street_name, street_type, street_suffix

locality name value `'the.rocks'`, would result in the single PID set (61560124) which corresponds to the original record. The location of this PID can then be look-up in the corresponding G-NAF geocode index. Table 2 shows the 23 attributes for which inverted indexes are built.

### 2.3 Additional Data Files

Additional information is used in the *Febrl* geocoding system during the preprocessing step to verify and correct (if possible) postcode and locality name values, and in the matching engine to enable searching for matches in neighbouring regions (postcodes and suburbs) if no exact match can be found.

Australia Post publishes a look-up table containing postcode and suburb information<sup>2</sup>, which can be used when processing the G-NAF locality files to verify and even correct wrong or missing postcodes and suburb names. For example, if a postcode is missing in a record, the Australia Post look-up table can be used to find the official postcode(s) of the suburb in this record, and if this is a unique postcode it can be safely imputed into the record. Similarly, missing suburb names can be imputed if they correspond to a unique postcode.

Other look-up tables are used to find *neighbouring* regions for postcodes and suburbs, i.e. for a given region these tables contain all its neighbours. These look-up tables are created using geographical data extracted from a commercial GIS system, and integrated into the *Febrl* geocode matching engine.

Look-up tables of both direct and indirect neighbours (i.e. neighbours of direct neighbours) are used in the geocode matching engine to find matches in addresses where no exact postcode or suburb match can be found. Experience shows that people often record different postcode or suburb values if a neighbouring postcode or suburb has a higher perceived social status (e.g. *'Double Bay'* and *'Edgecliff'*), or if they live close to the border of such regions.

<sup>2</sup> <http://www.auspost.com.au/postcodes/>

### 3 Geocode Matching Engine

*Febri's* geocode matching engine is based on the G-NAF inverted index data, and takes a rule-based approach to find an exact match or alternatively one or more approximate matches. Its input is a cleaned and standardised user record.

The matching engine tries to find an exact match first, but if none can be found it extends its search to neighbouring postcode and suburb regions. First direct neighbouring regions (level 1) are searched, then direct and indirect neighbouring regions (level 2), until either an exact match or a set of approximate matches can be found. In the latter case, either a weighted average location over all the found matches is returned, or a ranked (according to a likelihood rating) list of possible matches. The following steps explain in more detail (but still on a high conceptual level) how the matching engine works.

1. Find the set of address level matches (using street number and suffix) and the set of street level matches (using street name and type).
2. Find common matches between street and address levels (using set intersection).
3. Set the neighbour search level to 0 (no neighbouring regions are searched).
4. Find the locality match set (using locality name, qualifier and postcode) according to the current value of the neighbour search level. Postcode information is only used if no other locality information is available.
5. Find common matches between locality and address level, and between locality and street level (using set intersections).
6. If no matches between locality and address, and locality and street were found, increase the neighbour level (up to a maximum of 2) and jump back to step 4.
7. If matching records have been found, try to refine the match set using the postcode value (only if the postcode has not been used for the locality matches in step 4), as well as unit, flat and building (or property) information (if such information is available in the record).
8. If matches between street and address, or locality and address have been found, get their coordinates from the address geocode index. If only one match has been found, or if all found matches have the same location (this might be due to several G-NAF records corresponding to the same building) return the found location (longitude and latitude) together with an '**exact address match**' status. If more than one match with different locations have been found then calculate the average location and return it together with an '**average address match**' status.
9. If no address level match has been found use the street level match set. If only one match has been found or if all matches have the same location return the found location together with an '**exact street match**' status. If several street matches with different location were found return a '**many street match**' status and the list of found PIDs.
10. If no street level match has been found use the locality level match set. If only one match has been found or if all matches have the same location return the

**Table 3.** Matching results for geocoding 10,000 free-form LPI address records.

Match status	Number of records	Percentage
Exact address level match	7,288	72.87 %
Average address level match	213	2.13 %
Exact street level match	1,290	12.90 %
Many street level match	154	1.54 %
Exact locality level match	917	9.17 %
Many locality level match	135	1.35 %
No match	3	0.03 %

found location together with an 'exact locality match' status. If several locality matches with different location were found return a 'many locality match' status and the list of found PIDs.

11. If no match was found return a 'no match' status.

Geocoding of multiple addresses is an iterative process where each record is first cleaned and standardised, then geocoded and written into an output data set with coordinates and a match status added to each record.

## 4 Experimental Results

We have run experiments with geocoding various data sets. In this section we present initial results of geocoding a NSW *Land and Property Information* data set containing 10,000 randomly selected free-form addresses (from a data set containing around 2.7 million records). Table 3 shows the matching results. A total of 94.94% exact matches could be found at different levels. A closer analysis of the results showed that for 456 records no exact address match was found due to missing coordinates in the ADDRESS\_SITE\_GEOCODE file (i.e. our G-NAF data set did not have coordinates for these addresses). With better quality of future G-NAF releases we can therefore expect improved matching qualities.

Using a *SUN Enterprise 450* shared memory (SMP) server with four 480 MHz *Ultra-SPARC II* processors and 4 Giga Bytes of main memory, it took 23 minutes and 50 seconds to geocode the 10,000 address records, which is an average of 143 milli-seconds per record.

## 5 Conclusions and Future Work

In this paper we have described a geocoding system based on a geocoded national address file. We are currently evaluating and improving this system using raw uncleaned addresses taken from various administrative health related data sets. We are also planning to compare the accuracy of our geocoding system with commercial street level based GIS systems, and similar to [2] we expect more

accurate results. We are also fully integrating our geocoding system into the *Febrl* data linkage system [3, 7] and will publish it under an open source software license later this year.

Our main future efforts will be directed towards the refinement of the geocode matching engine to achieve more accurate matching results, as well as improving the performance of the matching engine (i.e. reducing the time needed to match a record). Three other areas of future work include:

- The *Febrl* standardisation routines currently return fields (or attributes) which are different from the ones available in G-NAF. This makes it necessary to map *Febrl* fields to G-NAF fields within the geocode matching engine. Better would be if the *Febrl* standardisation returns the same fields as the ones available in G-NAF, resulting in explicit field by field comparisons. We are planning to modify the necessary *Febrl* standardisation routines.
- Currently both the G-NAF preprocessing and indexing, as well as the geocode matching engine work in a sequential fashion only. Due to the large data files involved parallel processing becomes desirable. In the preprocessing step, the G-NAF data files can be processed independently or in a blocking fashion distributed over a number of processors, with only the final inverted indexes that need to be merged. Geocoding of a large user data file can easily be done in parallel as the cleaning, standardisation and matching of each record is independent from all others. An additional advantage of parallelisation is the increased amount of main memory available on many parallel platforms. We are planning to explore such parallelisation techniques and implement them into the *Febrl* system to allow faster geocoding of larger data sets. Additional performance improvements can be achieved by profiling and then replacing the core computational routines in the matching engine with C or C++ code.
- Geocoding uses identifying information (i.e. addresses) which raises privacy and confidentiality issues. Organisations that collect sensitive health data (e.g. cancer registries) cannot send their data to a geocoding service as this results in the loss of privacy for individuals involved. Methods are desirable which allow for privacy preserving geocoding of addresses. We aim to develop such methods based on techniques recently developed for blindfolded data linkage [5, 9, 10].

## Acknowledgements

This work is funded by the NSW Department of Health, Centre for Epidemiology and Research. The authors would like to thank David Horgan (student at the University of Queensland) who worked on a first version of this system while he was a summer student at the ANU. The authors also wish to thank PSMA for providing the G-NAF data files.

## References

1. Boulos, M.N.K.: Towards evidence-based, GIS-driven national spatial health information infrastructure and surveillance services in the United Kingdom. *International Journal of Health Geographics* 2004, 3:1. Available online at: <http://www.ij-healthgeographics.com/content/3/1/1>
2. Cayo, M.R. and Talbot, T.O.: Positional error in automated geocoding of residential addresses. *International Journal of Health Geographics* 2003, 2:10. Available online at: <http://www.ij-healthgeographics.com/content/2/1/10>
3. Christen, P., Churches, T. and Hegland, M.: A Parallel Open Source Data Linkage System. *Proceedings of the 8th PAKDD'04 (Pacific-Asia Conference on Knowledge Discovery and Data Mining)*, Sydney. Springer LNAI-3056, pp. 638–647, May 2004.
4. Churches, T., Christen, P., Lim, K. and Zhu, J.X.: Preparation of name and address data for record linkage using hidden Markov models. *BioMed Central Medical Informatics and Decision Making* 2002, 2:9, Dec. 2002. Available online at: <http://www.biomedcentral.com/1472-6947/2/9/>
5. Churches, T. and Christen, P.: Some methods for blindfolded record linkage. *BioMed Central Medical Informatics and Decision Making* 2004, 4:9, June 2004. Available online at: <http://www.biomedcentral.com/1472-6947/4/9/>
6. Ester, M., Kriegel, H.-P. and Sander, J.: *Spatial Data Mining: A Database Approach, Fifth Symposium on Large Spatial Databases (SSD'97)*. Springer LNCS 1262, pp. 48–66, 1997.
7. Freely extensible biomedical record linkage (Febri) project web page, URL: <http://datamining.anu.edu.au/linkage.html>
8. Fellegi, I. and Sunter, A.: *A Theory for Record Linkage*. Journal of the American Statistical Society, 1969.
9. Hok, P.: *Development of a Blind Geocoding System*. Honours thesis, Department of Computer Science, Australian National University, Canberra, November 2004.
10. O'Keefe, C.M., Yung, M., Gu, L. and Baxter, R.: *Privacy-Preserving Data Linkage Protocols*. *Proceedings of the Workshop on Privacy in the Electronic Society (WPES'04)*. Washington, DC, October 2004.
11. Paull, D.L.: *A geocoded National Address File for Australia: The G-NAF What, Why, Who and When?* PSMA Australia Limited, Griffith, ACT, Australia, 2003. Available online at: <http://www.g-naf.com.au/>
12. Rabiner, L.R.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, vol. 77, no. 2, Feb. 1989.
13. Rahm, E. and Do, H.H.: *Data Cleaning: Problems and Current Approaches*. IEEE Data Engineering Bulletin, 2000.
14. US Federal Geographic Data Committee. *Homeland Security and Geographic Information Systems – How GIS and mapping technology can save lives and protect property in post-September 11th America*. *Public Health GIS News and Information*, no. 52, pp. 21–23, May 2003.
15. Winkler, W.E.: *The State of Record Linkage and Current Research Problems*. RR99/03, US Bureau of the Census, 1999.



# Mining Optimal Item Packages using Mixed Integer Programming

N R Achuthan<sup>1</sup>

Raj P. Gopalan<sup>2</sup>

Amit Rudra<sup>3</sup>

<sup>1</sup>Department of Mathematics and Statistics  
Curtin University of Technology, Kent St, Bentley WA 6102, Australia  
archi@maths.curtin.edu.au

<sup>2</sup>Department of Computing  
Curtin University of Technology, Kent St, Bentley WA 6102, Australia  
raj@cs.curtin.edu.au

<sup>3</sup>School of Information Systems  
Curtin University of Technology, Kent St, Bentley WA 6102, Australia  
Amit.Rudra@cbs.curtin.edu.au

**Abstract.** Traditional methods for discovering frequent patterns from large databases are based on attributing equal weights to all items of the database. In the real world, managerial decisions are based on economic values attached to the item sets. In this paper, we introduce the concept of the value based frequent item packages problems. Furthermore, we provide a mixed integer linear programming (MILP) model for value based optimization problem in the context of transaction data. The problem discussed in this paper is to find an optimal set of item packages (or item sets making up the whole transaction) that returns maximum profit to the organization under some limited resources. The specification of this problem opens the way for applying existing and new MILP solution techniques to deal with a number of practical decision problems. The model has been implemented and tested with real life retail data. The test results are reported in the paper.

## 1. Introduction

With the proliferation of data available to an organization from day to day operations, the prospect of finding hidden nuggets of knowledge has greatly increased [19]. Traditional inventory systems help a retailer to keep track of what items are required to be stocked and when to replenish specific items. The issue these days is not just replenishing the stock on the shelves but also to group them according to their perceived association with items that attract the attention of the customer. Using past sales of frequent items and the association among them can be determined efficiently by current algorithms. The methods for finding the frequent patterns involve different types of partial enumeration schemes where all items are given equal importance. However, in most business environments, items are associated with varying values of price, cost, and profit. So, the relative importance of items differs significantly. Kleinberg et al. [1] noted that frequent patterns and association rules extracted from real life data would be of use to business organizations only if they are addressed

within the microeconomic context of the business. Brijs et al. [2] suggest that patterns in the data are interesting only to the extent to which they can be used in the decision making process of the enterprise. For example, the management of a supermarket may be interested in selecting packages of items that generate the maximum profit and requires physical storage space within certain limits. Another example is finding association rules where the items are most profitable or have the lowest margin.

Many such real-world problems can be expressed as optimization problems that maximize or minimize a real valued function. In this paper, we will focus on one such optimization problem in the context of transaction data and refer to it as **value based frequent item packages problem**. A package consists of items that are usually sold together. The aim is to find a set of items that can be sold as part of various packages to realize the maximum profit overall for the business.

Data mining research in the last decade has produced several efficient algorithms for association rule mining [3] [4] [5], with potential applications in financial data analysis, retail industry, telecommunications industry, and biomedical data analysis. However, literature on the use of these algorithms to solve real-world problems is limited [2]. Ali et al [6] reported the application of association rules to reducing fall-out in the processing of telecommunication service orders. They also used the technique to study associations between medical tests on patients. Viveros et al [7] applied data mining to health insurance data to discover unexpected relationships between services provided by physicians and to detect overpayments.

Most of the data mining algorithms developed for transaction data give equal importance for all the items. However, in a real business, not all the items are of equal value and many management decisions are made based on the money value associated with the items. The value may be in terms of the profit made or cost incurred or any other utility function defined on the items. Recent works by Aumann and Lindel [18] and Webb [17] discuss the quantitative aspects of association rules and tackle the problem using a rule based approach. More recently, Brij et al [2] developed a zero-one mathematical programming model for determining a subset of frequent item sets that account for total maximum profit from a pre-specified collection of frequent item sets with certain restrictions on the items selected. They used this model for the market basket analysis of a supermarket. Demiriz and Bennett [8] have successfully used similar optimization approaches for semi-supervised learning.

Mathematical programming has been applied as the basis for developing some of the traditional techniques of data mining such as classification, feature selection, support vector machines, and regression [9] [8]. However, these techniques do not address the value based business decision problems arising in the context of data mining and knowledge discovery. To the best of our knowledge, except for [2], mathematical modeling approach to classes of real world decision problems that integrate patterns discovered by data mining has not been reported so far. In this paper, we address this relatively unexplored research area and propose a new mathematical model for some classes of the value based frequent item packages problem. We contend that frequently occurring and profitable baskets are of greater importance to the retailer than subsets of transactions. The items that occur in a transaction can be packaged together or alternatively sold as individual items. We

consider the expected minimum revenue, minimum and maximum number of items in the optimal item packages, and storage constraint pertinent to a real life retailer.

The structure of the rest of this paper is as follows: In Section 2, we define relevant terms used in transaction data, frequent item sets and association rule mining. In Section 3, we consider a general version of an optimal item packages problem and present an integer linear programming formulation for the same. In Section 4, we illustrate the model by a sample profit optimal item packages problem; provide its MILP formulation and the result of processing it using commercial mathematical programming software (CPLEX). Finally, we conclude our paper in Section 5 providing pointers for further work.

## 2. Transaction Data - Notations and Definitions

**Transaction data** refers to information about transactions such as the purchases in a store, each purchase described by a transaction ID, customer ID, date of purchase, and a list of items and their prices. A web transaction log is another example in which each transaction may denote a user id, web page and time of access.

Let  $T$  denote the total number of transactions. Let  $I = \{1, 2, \dots, N\}$  denote the set of all potential items that may be included in any transaction and more precisely the items included in the  $t^{\text{th}}$  transaction may be denoted by  $I_t$ , a subset of  $I$ , where  $t$  ranges from 1 to  $T$ .

The **support**  $s$  of a subset  $X$  of the set  $I$  of items, is the percentage of transactions in which  $X$  occurs. A set of items  $X$  is a **frequent item set** if its support  $s$  is greater than or equal to a minimum support threshold specified by the user. An **association rule** is of the form  $X \Rightarrow Y$ , where  $X$  and  $Y$  are frequent item sets that do not have any item in common. We say that  $X \Rightarrow Y$  has **support**  $s$  if  $s\%$  of transactions includes all the items in  $X$  and  $Y$ , and **confidence**  $c$  if  $c\%$  of transactions containing the items of  $X$  also contains the items of  $Y$ . A valid association rule is one where the support  $s$  and the confidence  $c$  are above user-defined thresholds for support and confidence respectively. Association rules [10] [11] [12] identify the presence of any significant correlations in a given data set.

## 3. Optimal Item Packages Problem

Brijis et al [2] considered a market basket analysis problem for finding an optimal set of frequent item sets that returns the maximum profit and proposed a mixed integer linear programming (MILP) formulation of their problem. Their model proposed maximizing the profit function of frequent item set  $X$ , i.e.

$$\max \sum_{X \in L} M(X) * P_X - \sum_{i \in L} \text{Cost}_i * Q_i, \text{ where } L \text{ is the set of all frequent item sets}$$

$X$ ;  $M(X)$  is gross sales margin generated by  $X$ ; and  $P_X, Q_i \in \{0,1\}$  are decision

variables; subject to  $\sum_{i \in L} Q_i = ItemMax$ , where  $i$  is a basic item and  $ItemMax$  is the maximum threshold set for the number of items in  $X$ .

We generalize their problem to include different types of resource restrictions and develop an integer linear programming formulation for the same. The **Optimal Item Packages Problem (OIPP)** is to choose a set of frequent item sets or what we term as item packages, so as to maximize the total net profit subject to conditions on maximum storage space for selected items and minimum total revenue from the selected frequent item sets. Our formulation of the problem is much more flexible compared to Brij et al's [2], as the model can adapt to not only different resource restrictions but also to various bounds on the number of items in the final selection list. For example, it can specify the minimum and maximum number of elements in the final solution.

### 3.1 Motivation for OIPP

Often, in many real-life businesses, a transaction consists of a specific set of items as a package, facilitating a purchase. In such instances, both the number of items and the particular items forming the package are fixed. For example, while buying a car, a customer's choice may be made easier by having a number of fixed packages offered by the supplier. In some other businesses, it may not make sense to separate any item from a given package; e.g. medical procedures, travel packages etc.

Alternatively, a vendor may be interested in finding out from previous sales as to which, if any, set of items exist that could be offered as a package. This packaging of items (or products) could potentially offer him certain amount of profit under a number of resource constraints. For instance, the resource constraints could be available stocking space, budget (minimum cost or maximum profit), quantity (that needs to be sold) etc. He may be further interested in doing a sensitivity analysis as to how far the resources can be stretched while the given solution remains optimal. Again, in another instance, the vendor may like to see how a change in a certain resource affects his profitability (for example, if he is able to organize a little more space for storage or invest a little more money). For a travel bureau, a constraint could be time-oriented resources (like, a travel consultant's time),

#### OIPP:

For a given database  $\{a_{it} : 1 \leq i \leq N, 1 \leq t \leq T\}$ , let  $\{X_j : 1 \leq j \leq k\}$  be a pre-specified list of  $k$  frequent item sets. Let  $f_j$  and  $n_j$  respectively denote the number of transactions that exactly include  $X_j$  (i.e.  $f_j = |\{t : I_t = X_j\}|$ ) and the number of items in  $X_j$ ,  $1 \leq j \leq k$ . Let  $p_j$  denote the revenue made by the frequent item set  $X_j$  whenever  $X_j$  forms a transaction. Let  $c_i$  denote the cost incurred (per unit) while selecting item  $i$ ,  $1 \leq i \leq N$ . Let  $s_i$  denote the storage space (in appropriate units) required per unit for item  $i$  whenever the item is selected. Furthermore, let  $S$  denote the total available storage space. Find, a subset  $\hat{I}$  of  $\{i : 1 \leq i \leq N\}$  and a subset  $F$  of the set of frequent item sets  $\{X_j : 1 \leq j \leq k\}$  such that they satisfy the following properties:

1. The number of items in  $\hat{I}$  is bounded below and above by positive integers  $N_L$  and  $N_U$  respectively;
2. A frequent item set  $X_j$  is selected in  $F$  if and only if  $X_j$  is covered by  $\hat{I}$ , that is,  $X_j \subseteq \hat{I}$ ;
3. The total storage space required for the selected items of  $\hat{I}$  does not exceed the available space of  $S$  units;
4. The total revenue made by frequent item sets of  $F$  is at least  $\text{Minrev}$ ;
5. The net profit (total revenue – the total cost) is maximized.

We now provide an MILP model for the OIPP described above. Let  $y_i$  denote the 0-1 decision variable that assumes value 1 whenever item  $i$  is chosen. Let  $z_j$  denote the 0-1 decision variable that assumes value 1 whenever the frequent item set  $X_j$  is covered by the set of selected items, that is, by the set of items  $\{i: y_i = 1\}$ .

$$(6) \text{ Lower and upper bound constraints: } N_L \leq \sum_{i=1}^N y_i \leq N_U$$

$$(7) \text{ Occurrence constraint of } X_j: \sum_{i \in X_j} y_i - n_j z_j \geq 0, \quad 1 \leq j \leq k$$

$$(8) \text{ Item storage space constraint: } \sum_{j=1}^k \left( \sum_{i=1}^N b_{ij} s_i \right) f_j z_j \leq S$$

$$(9) \text{ Lower bound constraint on revenue: } \sum_{j=1}^k p_j f_j z_j \geq \text{Minrev}$$

$$(10) \text{ Restrictions on variables: } y_i = 0 \text{ or } 1, z_j = 0 \text{ or } 1$$

$$(11) \text{ Objective function: } \text{Maximize } \sum_{j=1}^k p_j f_j z_j - \sum_{j=1}^k \left( \sum_{i=1}^N b_{ij} c_i \right) f_j z_j$$

In this value based frequent item set problem the input information regarding  $X_1, \dots, X_k$ ,  $p_j$ ,  $f_j$ ,  $s_i$  and  $c_i$  must be extracted through data mining of frequent item sets. For the above model (6) – (11), the constraints and the objective function may be validated as follows:

Let the set of selected items to cover all the selected frequent item sets be denoted by

$$\hat{I} = \{i: y_i = 1\}. \text{ It is easy to see that } |\hat{I}| = \sum_{i=1}^N y_i \text{ and the constraint (6) provides the}$$

lower and upper bound restrictions on this number. The number of items common to

the set  $\hat{I}$  and the frequent item set  $X_j$  is given by  $\sum_{i \in X_j} y_i$ . Whenever  $\sum_{i \in X_j} y_i = |X_j| =$

$n_j$ , the set  $\hat{I}$  covers the frequent item set  $X_j$ . The constraint (7) ensures that the decision variable  $z_j$  is 1 if and only if the frequent item set  $X_j$  is covered by  $\hat{I}$ . In this case note that  $F = \{X_j : z_j = 1\}$  is the collection of frequent item sets selected. The

storage space required by an item of the selected item set  $X_j$  is  $\sum_{j=1}^k b_{ij} s_i f_j z_j$ ,

where  $b_{ij}$  is a known constant that takes value 1 or 0 according as item  $i$  is in item set  $X_j$  or not for  $1 \leq i \leq N$ , and  $1 \leq j \leq k$ . The constraint (8) expresses the upper bound restriction on the available storage space viz.  $S$ . The contribution made by the frequent item set  $X_j$  to the profit may be expressed as  $p_j f_j z_j$  where  $z_j = 1$  if and only if  $X_j$  is covered by the set  $\hat{I}$ . The constraint (9) ensures a minimum revenue contribution from the set of all covered frequent item sets. The constraints of (10) express the 0-1 restrictions of the decision variables  $y_i$  and  $z_j$ . The objective function in (11) maximizes the total profit contribution expressed as the total net revenue.

## 4. Experimental Results

To verify our MILP formulation of the OIPP, we implemented and experimentally tested our model with real life market transaction data obtained from a Belgian retail store [16]. The dataset (retail.txt) stores five months of transaction data collected over four separate periods.

Retail data characteristics:

Total number of transactions	88,163
Item ID range	1- 16470
Number of items (N)	3,151
Total number of customers	5,133
Average basket size	13
Data collection period	5 months total (in four separate periods)

For further details of the data refer to [16]. Since not all characteristics of the data are publicly available (presumed to be confidential), we supplemented them with values for such fields as storage space required per item ( $s_i$ ), revenue from selling item package  $X_j$  ( $p_j$ ) and cost attributed to item  $i$  ( $c_i$ ).

### 4.1 Data preparation stages

As discussed in section 3, before building the MILP model of the market data we need to know the data characteristics. Therefore, the data is preprocessed using the following steps to prepare it for input to the mathematical programming software:

1. Each transaction record is organized as an ascending sequence of item Ids;
2. A count of the number of items ( $n_j$ ) in each transaction is inserted as the first field of the record;
3. The records in the database are then sorted in ascending order according to the count of items and then by the item Ids as minor keys;
4. Finally, identical transactions are counted to obtain their frequencies ( $f_j$ );
5. This final dataset is fed to a program (createLP) which builds the MILP model corresponding to the current problem.

This model is then submitted to a mathematical programming application to be solved as an MILP with binary integer variables ( $y_i$ 's and  $z_j$ 's).

[We used C++ programs (for steps 1 – 5 above) to process the input retail market basket dataset and produced the output in appropriate format. As our MILP formulation assumes data mining activities as a pre-step, discussions regarding the preprocessing done by these programs are unnecessary.]

#### 4.2 Sample optimal package selection problem

To help explain our methodology, we use an example problem and work it through the different stages of finding the optimal profit from the given dataset. Consider the following dataset consisting of 5 sales transactions involving 7 items.

$$X_1 = \{7\}, X_2 = \{1, 2\}, X_3 = \{5, 6\}, X_4 = \{12, 13\} \text{ and } X_5 = \{2, 6, 12, 13\}$$

Table 1 below, shows the characteristics of various items ( $sp_i$  – selling price,  $prof_i$  – profit per unit); while Table 2 presents the details of each item package.

Item	1	2	5	6	7	12	13
$s_i$	0.2	0.3	0.25	0.3	0.15	0.3	0.2
$c_i$	2.5	3.1	4.5	3.7	2.1	3.5	2.5
$s.p._i$	3.2	3.9	6.7	4.9	2.6	4.0	3.1
$prof_i$	0.7	0.8	2.2	1.2	0.5	0.5	0.6

**Table 1.** Characteristics of items in the sample dataset

Count of items ( $n_j$ )	Number of transactions ( $f_j$ )	Item package ( $X_j$ )	Package revenue ( $p_j * f_j$ )	Package storage ( $\sum s_i * f_j$ )	Package cost ( $\sum c_i * f_j$ )
1	200	7	520	30	420
2	231	1 2	1640.1	115.5	1293.6
2	34	5 6	394.4	18.7	278.8
2	341	12 13	2421	170.5	2046
6	11	2 6 12 13	174.9	12.1	140.8

**Table 2.** Processed sample dataset for creating the MILP model

The first column shows the number of items in the packages viz.  $n_j$ ; the second shows the frequency ( $f_j$ ); while the third shows the individual items that make up each item package. The last three columns show the computed aggregates for each package.

The createLP program, outlined in step 5 above, processes the formatted dataset (steps 1-4) and produces the corresponding MILP model (Fig. 1) to the sample dataset. This model is then solved using CPLEX, a commercial package for solving the all kind of linear programs. Fig. 2 presents the output from the package.

```

\Problem name: sample.lp

Maximize

    100z1 + 346.5z2 + 115.6z3 + 375.1z4 + 34.1z5

Subject To


    -1z1 +y7 >= 0
    -2z2 +y1 +y2 >= 0
    -2z3 +y5 +y6 >= 0
    -2z4 +y13 +y12 >= 0
    -4z5 +y2 +y6 +y12 +y13 >= 0


    30z1 + 115.5z2 + 18.7z3 + 170.5z4 + 12.1z5 <= 100
    520z1 + 1640.1z2 + 394.4z3 + 2421.1z4 + 174.9z5 >= 600


    y1 +y2 +y5 +y6 +y7 +y12 +y13 >= 5
    y1 +y2 +y5 +y6 +y7 +y12 +y13 <= 10

Binaries
    z1 z2 z3 z4 z5
    y1 y2 y5 y6 y7 y12 y13

End
    
```

*Max. storage  
constraint*  


 *Min. revenue  
constraint*

 *lower & upper  
bounds*

**Fig. 1. Sample problem sample.lp**

```

Integer optimal solution: Objective = 2.4970000000e+002
Solution time = 0.03 sec. Iterations = 0 Nodes = 0

CPLEX> dis sol var -
Variable Name      Solution Value
z1                  1.000000
z3                  1.000000
z5                  1.000000
y7                  1.000000
y2                  1.000000
y5                  1.000000
y6                  1.000000
y13                 1.000000
y12                 1.000000

All other variables in the range 1-12 are zero.
    
```

**Fig. 2. Solution of sample MILP using CPLEX**

We notice (Fig. 2) that the optimal value i.e. the maximal profit, obtained under the given constraints of 100 units of storage space and satisfying the minimum revenue of

\$600 is \$249.70. The three best item packages to stock are  $X_1$ ,  $X_3$  and  $X_5$  which correspond to the binary decision variables  $z_1$ ,  $z_3$  and  $z_5$  respectively. Further, the particular items in the optimal set to store are 7, 2, 5, 6, 12 and 13 (corresponding to the decision variables  $y_7$   $y_2$   $y_5$   $y_6$   $y_{12}$   $y_{13}$ ). The remaining item packages and items do not participate in the optimal solution.

We now present the results from the retail dataset as described at the beginning of the section. Given a certain maximum storage space, the retailer might like to find out the optimum profit (and item packages) against a maximum number of items to be put on the shelves. He might also be curious as to how the profit varies if he is able to acquire more storage space. To show how easily this can be achieved using our MILP formulation, we varied the values for  $S$ , the maximum storage space parameter, from 1000 to 4000 and varied the upper limit for the number of items to be shelved i.e.  $N_U$  from 20 to 500. The resulting MILP was then submitted to CPLEX 9.0 to calculate the value of the net profit function ( $z$ ). Table 3 shows the effect of changing the maximum number of items ( $N_U$ ) has on the objective.

4000									
Nu	20	50	100	150	200	300	400	450	500
profit	22,697	27,996	32,323	34,646	36,320	39,281	39,384	39,381	39,384
time	0.24	0.24	0.23	0.26	0.23	0.24	0.22	0.12	0.11

3000									
Nu	20	50	100	150	200	300	400	450	500
profit	22,697	27,996	32,323	34,646	36,320	39,281	39,281	39,328	39,328
time	0.25	0.22	0.23	0.26	0.22	0.24	0.22	0.35	0.34

2000									
Nu	20	50	100	150	200	300	400	450	500
profit	22,697	27,996	32,323	34,646	36,320	38,315	38,423	38,423	38,423
time	0.25	0.22	0.23	0.26	0.21	0.22	0.23	0.23	0.23

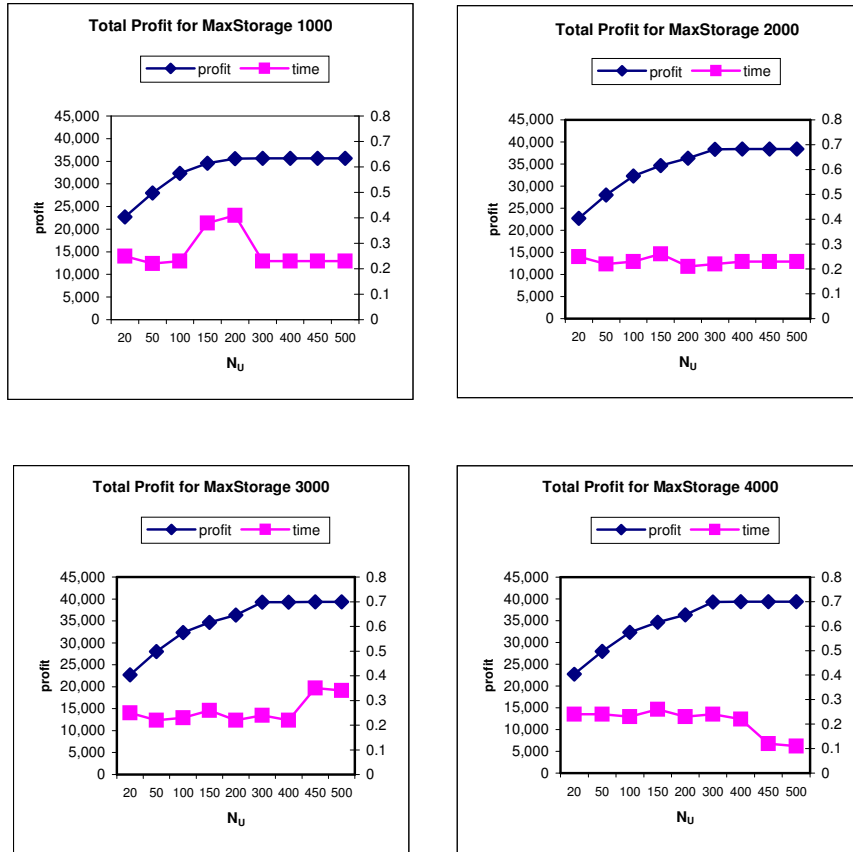
  

1000									
Nu	20	50	100	150	200	300	400	450	500
profit	22,697	27,996	32,323	34,556	35,564	35,636	35,636	35,636	35,636
time	0.25	0.22	0.23	0.38	0.41	0.23	0.23	0.23	0.23

Table 3. Comparative values of the profit function as the number of items to be stocked and storage available are varied.

We then chart (Figure 3) the observations to visualize the effects of max. storage and  $N_u$  on the value of the objective,  $z$ . We observe that while increasing the number of items does increase the net profit quite substantially, after a certain stage the rate or amount of change in the same is not significant, eventually peaking and remaining so in spite of increasing resources (storage space or number of items stored). This observation could be of value to the retailer as he can clearly visualize the expected

changes in profit by changing certain parameters as need be. Similarly, one can study the effect of varying the limits of other resources and study their effects on the profitability function.



**Fig. 3 Effect of varying max storage and max number of items on the objective**

For our experiments, we used an AMD Athlon XP2100 PC with a CPU clock of 2.1 GHz having 512 MB of RAM running Windows 2000. Our experiments show very encouraging results as all of them are achieved in a sub-second response time. This proves that our method of solving such problems is very much viable.

## 5. Conclusions

In this paper, we have introduced a general class of problems called the value based optimal item package problem that can support real world business decisions using

data mining. The solutions to these problems require the combination of mathematical modeling with data mining and knowledge discovery from large transaction data. We formulated a generic problem using the mixed integer linear programming model and implemented it using real life transactional data from a retail store. Our specification provides scope for using a large number of methodologies available in the literature to solve the value based frequent item set problems.

It is well known that the general integer linear programming problem is NP hard. In addition, in many practical applications of the frequent item set problem, the parameters like  $N$ , the number of items and  $T$ , the number of transactions in the data base may be very large. When  $N$  and  $T$  are not very large, we can use some of the standard commercial software products such as CPLEX to solve the model proposed in this paper. Furthermore, future research can be focused on developing specially designed branch and cut algorithms [13] [14] [15], branch and price algorithms and/or efficient heuristics and probabilistic methods to solve our MILP formulations of these models. When  $N$  and  $T$  are large, the future research can explore the possibility of solving these models restricted to some random samples drawn from the database and developing methods of estimating the required information.

## References

1. Kleinberg, J., Papadimitriou, C., Raghavan, P.: A Microeconomic View of Data Mining. *Data Mining and Knowledge Discovery*. **2** (1998) 311-324
2. Brijs, T., Swinnen, G., Vanhoof, K., Wets, G.: Building an Association Rules Framework to Improve Product Assortment Decisions. *Data Mining and Knowledge Discovery*. **8** (2004) 7-23
3. Gopalan, R.P., Suchahyo, Y.G.: High Performance Frequent Patterns Extraction using Compressed FP-Tree. *Proceedings of SIAM International Workshop on High Performance and Distributed Mining (HPDM04)*, Orlando, USA (2004)
4. Han, J., Pei, J., Yin, Y.: Mining Frequent Patterns without Candidate Generation. *Proceedings of ACM SIGMOD*, Dallas, TX (2000)
5. Liu, J., Pan, Y., Wang, K., Han, J.: Mining Frequent Item Sets by Opportunistic Projection. *Proceedings of ACM SIGKDD*, Edmonton, Alberta, Canada (2002)
6. Ali, K., Manganaris, S., Srikant, R.: Partial Classification using Association Rules. *Proceedings of KDD-97*, Newport Beach, California (1997)
7. Viveros, M.S., Nearhos, J.P., Rothman, M.J.: Applying Data Mining Techniques to a Health Insurance Information System. *Proceedings of VLDB-96*, Bombay, India, 1996.
8. Demiriz, A., Bennett, K.P.: Optimization Approaches to Semi-Supervised Learning. In *Complementarity: Applications, Algorithms and Extensions*. Kluwer Academic Publishers, Boston (2001) 121-141
9. Bradley, P., Gehrke, J., Ramakrishnan, R., Srikant, R.: Scaling Mining Algorithms to Large Databases. *Communications of the ACM*. **45** (2002) 38-43
10. Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R.: *Advances in Knowledge Discovery and Data Mining*. MIT Press, Cambridge, MA (1996)
11. Han, J., Kamber, M.: *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, San Francisco (2001)
12. Hand, D., Mannila, H., Smyth, P.: *Principles of Data Mining*. MIT Press, Cambridge, MA (2001)

13. Achuthan, N.R., Caccetta, L., Hill, S.P.: A New Subtour Elimination Constraint for the Vehicle Routing Problem. *E.J.O.R.* **91** (1996) 573-586
14. Achuthan, N.R., Caccetta, L., Hill, S.P.: Capacitated Vehicle Routing Problem: Some New Cutting Planes. *Asia-Pacific Journal of Operational Research*. **15** (1998) 109-123
15. Achuthan, N.R., Caccetta, L., Hill, S.P.: An Improved Branch and Cut Algorithm for the Capacitated Vehicle Routing Problem. *Transportation Science*. **37** (2003) 153-169.
16. Brijs T., Swinnen G., Vanhoof K., and Wets G. The Use of Association Rules for Product Assortment Decisions: A Case Study, in: *Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining*, San Diego (USA), August 15-18, (1999) 254-260.
17. Webb, G. Discovering Associations with Numeric Variables. *Proceedings of the Knowledge Discovery in Databases (KDD 01)*, San Francisco (USA), (2001) 383-388.
18. Aumann, Y., Lindell, Y. A Statistical Theory for Quantitative Association Rules. *Proceedings of the Knowledge Discovery in Databases (KDD 99)*, San Francisco (USA), (1999) 262-270.
19. Marakas, G. M. *Modern Data Warehousing, Mining and Visualization*. Prentice Hall, Upper Saddle River, New Jersey (USA). (2003).

# Decision Theoretic Fusion Framework for Actionability Using Data Mining On an Embedded System

\*Heungkyu Lee, \*\*Hanseok Ko

\*Dept. of Visual Information Processing, \*\*Dept. of Electronics and Computer Engineering  
Korea University, Seoul, Korea  
hklee@ispl.korea.ac.kr, hsko@korea.ac.kr

**Abstract.** This paper proposes a decision theoretic fusion framework for actionability using data mining techniques in an embedded car navigation system. An embedded system, having limited resources cannot manage the abundant information in the database. Thus, the proposed system stores and manages only multiple level-of-abstraction in the database to resolve the problem of resource limitations, and then represents the information received from the Web via the wireless network after connecting a communication channel with the data mining server. To do this, we propose a decision theoretic fusion framework that includes the multiple level-of-abstraction approach combining multiple-level association rules and the summary table, as well as an active interaction rule generation algorithm for actionability in an embedded car navigation system. In addition, it includes the sensory and data fusion level rule extraction algorithm to cope with simultaneous events occurring from multi-modal interface. The proposed framework can make interactive data mining flexible, effective, and instantaneous in extracting the proper action item.

**Keywords:** Data mining, Embedded data mining, and Speech interactive approach.

## 1 Introduction

As detailed and accurate data are accumulated and stored in databases at various stages, the large amounts of data in databases makes it almost impractical to manually analyze them for valuable information. Thus, the need for automated analysis and discovery tools to extract useful knowledge from huge amounts of raw data has been urgent. To cope with this problem, data mining methodologies are emerging as efficient tools in realizing the above objectives. Data mining [1][2][3] is the process of extracting previously unknown information in the form of patterns, trends, and structures from large quantities of data. These methodologies are being used in many fields, such as financial, business, medical, manufacturing and production, scientific domains, and the World Wide Web (WWW). Especially, autonomous decision-

making process using a data mining approach has been useful in various fields for sourcing efficient and reliable information [4][5].

In addition, as computer and scientific technologies have improved recently, small size handheld mobile devices such as PDAs, mobile phones, and Auto PCs have been used in various fields of mobile computing and Telexistence technologies more and more. The need to utilize a variety of service applications such as car navigation, MP3/WAV player, car maintenance program, and information center solution connecting to server, on these devices is increasing. However, an embedded hardware system has limited resources that are not enough to handle the large amounts of data, and analyze them. Thus, an embedded technique to resolve this problem is required.

To cope with this problem, we propose a decision theoretic fusion framework that includes the multiple level-of-abstraction approach which combines multiple-level association rules and a summary table as well as active interaction rule generation algorithm for actionability in an embedded car navigation system. In addition, it includes the sensory and data fusion level rule extraction algorithm to cope with simultaneous events occurring from multi-modal interfacing. This embedded system is connected to the data mining server based on the web in order to extract and access the rules and data. This is because the Web not only contains a huge amount of information, but also can provide a powerful infrastructure for communication and information sharing [6][7]. With this data mining server, the proposed system can provide an efficient data representative service as well as actionability to present interactive methods without processing the raw data.

The proposed system is applied to command, control, communication, and intelligent car navigation systems. This provides an efficient speech interactive agent (SIA) rendering smooth car navigation by employing a conversational tool; embedded automatic speech recognition, embedded text-to-speech, and distributed speech recognition modules, all the while enabling safe driving. The embedded car navigation system is extended to provide a user-friendly service and interactive capability by using the conversational tools. The system can reveal the status of the system and its scheduled jobs by actively using the active interaction rule generation algorithm. This is due to the fact that the driver has an access pattern about specific applications that are frequently used. In addition, the information about traffic, weather, news, daily schedules, and car management can provide valuable information to the driver as well as decision-making advice on what action the proposed system should take. Using such information, the speech interactive agent provides efficient interactive methods to operate for the required events.

First, this system uses sensory fusion rules in order to combine multiple events simultaneously occurring from multi-modal sensors such as push-to-talk, remote controller, touch screen, mute, hands-free, external buttons, and application events received from multimedia service applications in the embedded client system. Second, the data fusion framework is provided by using the features extracted from sensory fusion rules. At this time, user access patterns occurring by user driven events operate a specific service application, and are mined and stored in databases on an embedded system for certain periods. This feature provides the means to decide a specific action. The proposed system can connect the Internet server using a CDMA 200 terminal to represent large amounts of information. However, an embedded system has a small sized memory that has not enough space to store a lot of

has a small sized memory that has not enough space to store a lot of information. To resolve this problem, the multiple level-of-abstraction approach for the multiple-level association rules is applied.

The content of this paper is as follows. The design concept of the proposed system is presented in Section 2. In Section 3, we describe the data mining methodologies based on the decision theoretic fusion framework for actionability. Finally, in Section 4, we provide discussions and conclusive remarks.

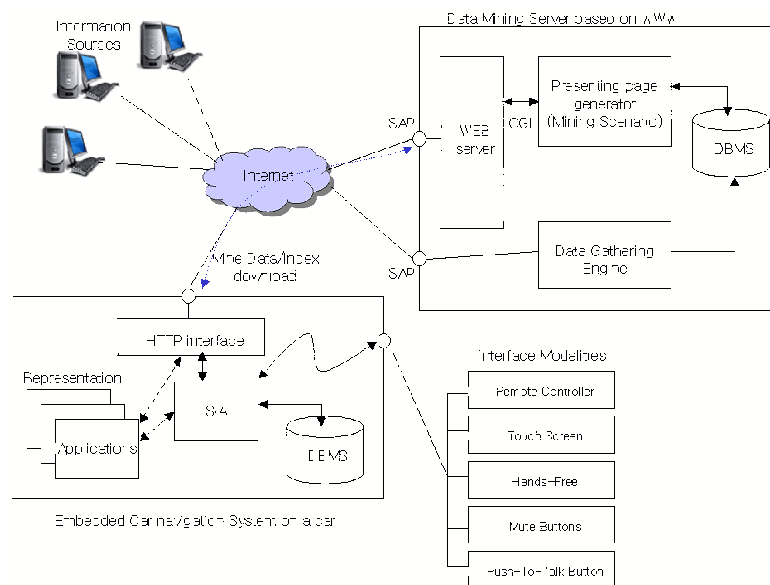


Fig 1. System architecture overview

## 2 Architecture of Embedded Car Navigation System

### 2.1 System Overview

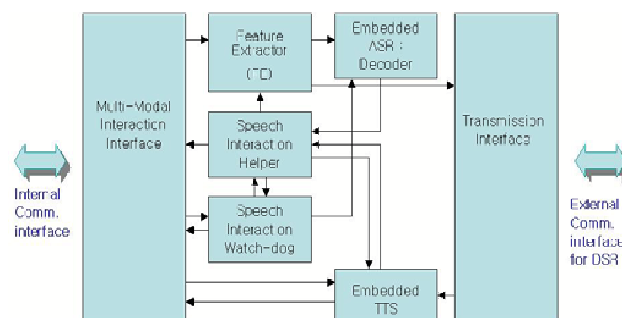
The embedded car navigation system provides the various embedded service applications on a car as well as networked service applications via a wireless network using the CDMA 2000 terminal as shown in Figure 1. In our proposed system, we include the interactive techniques using speech interactive agent to provide a speech interaction method as an intelligent interface between human and machine. The speech interactive agent plays a role in combining and processing the information from interface modalities as well as in communicating with the data mining server to provide useful information to the user. This system needs a database to store some

valuable information and manage some information. Such an embedded system has limited resources. To resolve this problem, this system stores and manages the multiple level-of-abstraction in the database. The multiple level-of-abstraction information is downloaded, and updated from the data mining server using HTTP (Hyper-Text Transfer Protocol). In addition, this system can manage the user's access patterns providing the user used the service for a certain period. By using this information, the speech interactive agent can speak to the user when the system is first switched on at the start of the day, and the scheduled job should be executed. This information is also managed in the database by using multiple level-of-abstraction.

## 2.2 Speech Interactive Agent

Conversation is one of the most important factors that facilitate dynamic knowledge interaction. People can have a conversation with a conversational agent that talks with people by using the eASR and eTTS [8] as a combined unit. The speech interaction agent, as a conversational agent [9][10], carries out command and control tasks while interacting with the driver according to the given scenarios on the car navigation system, as in our previous work [11].

As a problem-solving paradigm, the fusion process model using the functional evaluation stage is employed [12]. Although the car navigation system is deterministic, the use of multiple input sensors makes the system complex to cope with various situations. The proposed speech agent is decomposed into three separate processes; composition process of sensory sources, speech signal processing process and decision-making process. As shown in Figure 2, the composition process of sensory sources plays a role in combining input requests and guiding the next-step. The speech signal processing process provides a means of speech interaction using speech recognition and text-to-speech functions. The decision-making process provides a user-friendly interfacing mode using a speech interaction helper function as well as a self-diagnosis function using a speech interaction watch-dog module.



**Fig 2.** Speech Interactive Agent (SIA) block diagram

The speech recognition system is classified into the embedded ASR and distributed speech recognition (DSR) system that is used via the wireless network, using a CDMA 2000 terminal. Thus, the feature extractor based on ETSI v1.1.2 has the

front-end role of passing the mel-cepstral features to eASR or DSR according to the scenarios without communicating between the speech agent and the application process. The eTTS utters the information when the event is requested by the user and application programs. The "speech interaction helper" provides helper scenarios to the user when a recognition error occurs or an out-of-vocabulary is encountered. The "watch-dog" function monitors the service situation and status of the eASR/eTTS in order to cope with the exception-handling error which can occur when a user pushes the external buttons during the service interval.

### 3 Decision Theoretic Fusion Framework

#### 3.1 Sensory Fusion Rule

To perform the requests for speech interaction, firstly the sensory fusion model can be expressed by

$$Y_i = f(O/K, Y_{i-1}) \quad (1)$$

where  $i$  is a number of processing results,  $O$  is a observable sensory input,  $K$  is a domain knowledge,  $Y_{i-1}$  is status information being processed from a previous time and  $f()$  is the sensory fusion function to combine the sensory inputs and then control the current requests given the previous situation. The observable sensor input,  $O$  is expressed by

$$O = g_1(Mute) \cdot g_2(HF) \cdot g_3(R) \cdot g_4(Ptt) \cdot \prod_{i=0}^k g_5(E_i) \quad (2)$$

where  $M$  is a mute, HF is a hands-free,  $R$  is a remote controller,  $Ptt$  is a push-to-talk,  $E$  is a event created by service applications, and  $k$  is a number of applications being run simultaneously. Each input is independent each other as well as processed parallelly. The variable,  $g()$  is a function to observe and detect the sensor input. While a sensory input between  $g_1$  and  $g_4$  is a direct input from a sensor,  $g_5$  is a transmitted input from application programs via the inter-process communication. The sensory inputs can happen simultaneously. However, for the action to be performed promptly it is always one function that is most suitable in a given situation. This is due to the fact that the hardware resource has limitations, and the system can provide the robustness, consistency and efficiency in using a service. Thus, the fusing function,  $f()$  should be considered with respect to service quality and usability. In this paper, we apply the rule based decision function as a fusion function of respective inputs. In equation (1),  $K$  is a domain specific knowledge to provide combing rules as shown in Table 1. The given rule is decided by considering the service capability, priority and resource limitations, etc. Decision categories are composed of five decision rules.

**Table 1.** The negotiation rule table according to the priority control

Current State Previous State	eASR is requested	Application TTS is requested	CNS TTS is requested	Hands-Free Button pushed	Mute Button pushed
Hands-Free button enable	Disabled	Disabled	Enabled	Not applicable	Not applicable
Mute button enable	Enabled	Disabled	Enabled	Not applicable	Not applicable
eASR running	Previous eASR exits and new eASR runs	Previous eASR exits and eTTS starts	eASR runs continuously and CNS TTS starts	eASR exits	eASR exits
Application eTTS running	Previous eTTS stops and eASR runs	Previous eTTS stops and new eTTS starts	Application eTTS pauses and CNS TTS starts	eTTS stops	eTTS stops
CNS eTTS running	CNS eTTS starts and eASR runs	Previous CNS eTTS finishes and then application eTTS starts	Previous CNS eTTS stops and new CNS eTTS starts	Don't care	Don't care

### 3.2 Data Fusion Rules from Interface Modalities

When given the sensory fusion result, the speech agent can decide the action to be performed. Next, the data fusion model for speech interaction can be expressed by

$$Z = H_i(O_i) \cdot I(P) \cdot J(Y), \quad i = 1, \dots, 3 \quad (3)$$

$$H_i(O_i) = h_i(O_i / M_i), \quad i = 1, \dots, 3 \quad (4)$$

where  $i$  is the number of speech interaction tools and  $H_i(O_i)$  is a speech interaction tool; 1)embedded speech recognition, 2)distributed speech recognition 3)text-to-speech. Thus, the variable,  $O_1$  and  $O_2$  are speech sampling data and  $O_3$  is text data. Thus,  $H_i(O_i)$  is decomposed as follows.

$$\begin{aligned} H_1(O_1) &= h_1(O_1 / M_1) \\ &\cong W_k = \arg \max_j L(O / W_j) \end{aligned} \quad (5)$$

where  $h_1(O_1)$  is a pattern recognizer using the maximum a posteriori (MAP) decision rule to find the most likely sequence of words.

$$H_2(O_2) = h_2(O_2 / M_2) = h_2(O_2) \quad (6)$$

where  $h_2(O_2)$  is a front-end feature extractor to pass the speech features into the back-end distributed speech recognition server.

$$H_3(O_3) = h_3(O_3 / M_3) \quad (7)$$

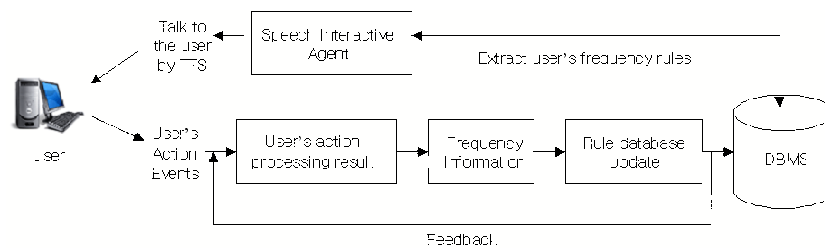
where  $h_3(O_3)$  is a speech synthesizer function to read the sentences.

$J(Y)$  is a selecting function to choose a speech interaction tool. The currently selected speech module is just enabled. The variable,  $M_i$  is a given specific domain knowledge.  $M_1$  is an acoustic model to recognize the word,  $M_2$  is not used and  $M_3$  is TTS DB. The variable,  $P$  is procedural knowledge to provide a user-friendly service such as a helper function.  $I(P)$  is a function to guide the service scenario according to the results of the speech interaction tool.

As a result,  $Z$  is an action to be performed sequentially. The final decision-making,  $Z(t)$  represents the user's history to be processed when the decision is stored for a long period of time. This can provide the statistical information when the user frequently utilizes a specific function.

### 3.3 Active Interaction Rule Generation

Users may interact at various service stages and domain knowledge may be used in the form of a higher-level specification of the model, or at a more detailed level. In our system, the speech interactive agent interacts with users using a conversational tool. This user interaction information is applied to data mining which is inherently an interactive and iterative process. This is due to the fact that the user has repeated patterns that he or she frequently uses on specific applications with the car navigation system. By using this information, the speech interactive agent asks the user whether the user wants to perform a specific task, which is the statistical information to be stored and estimated for a period of time according to the procedure in Figure 3. In addition, the speech interactive agent can start a music player automatically according to the days' weather broadcasts if the system has not been used for a long time. This function can be set on or off manually on an application by a user. To obtain some information for specific tasks, the speech interactive agent downloads and updates the mined data from the data mining server via the wireless internet.



**Fig 3.** Active interaction procedure using the user's frequency rule.

To extract the features for data mining, the rough set theory [1] is applied. By using the rough set theory, a decision rule induction from an attribute value table is done. The feature extraction algorithm can generate multiple feature sets (reducts). These feature sets are used for predicting the user's action with the primary decision-making algorithm and confirmation algorithm. The primary decision-making algorithm compares the feature values of objects with decision rules. If a matching criterion is found, the decision rule for action of the speech interactive agent is assigned to the specific job. However, the user may not require the specific task to be performed because of lack of confidence if the user is distracted at that time. Thus, the confirmation algorithm is applied using speech interaction tools; speech recognition and text-to-speech. When the user just says "yes", the action is performed according to the rule of the decision-making algorithm.

We select five features, F1-F5. F1 is the indicator to notify whether the system is in the sleep mode or not. F2 is the indicator to notify whether the object (application ID) is one reserved at the scheduled time or not. F3 is the reserved time if the F2 is set to 1. F4 is the frequency rate when the object is used for some time. F5 is the priority of that application. Table 2 includes 3 decision rules generated with the rule extraction algorithm. The decision rules are followed continually when the F1 is just set to 1. If the matching criterion is met in the next decision rules, decision rule is set to N, which is the object number to be performed by the speech interactive agent. Table 3 depicts a sample data set. When F1 is 1, F2 is 1, and F3 of the object 2 is on time to be executed, the decision rule, D is set to 2. Thus, the object 2 is selected as the one that can be executed. If F3 does not notify by a scheduled time, the decision rule, D is set to 5 because the object number 5 has the highest priority,  $F5=1$ .

**Table 2.** Decision rules for the action

Decision rule 1. IF (F1 = 0) THEN (D = 0)
Decision rule 2. IF (F2 = 1) AND (F3 = NOW) THEN (D = N)
Decision rule 3. IF (F4 = 1) AND (F5 = 1) THEN (D = N)

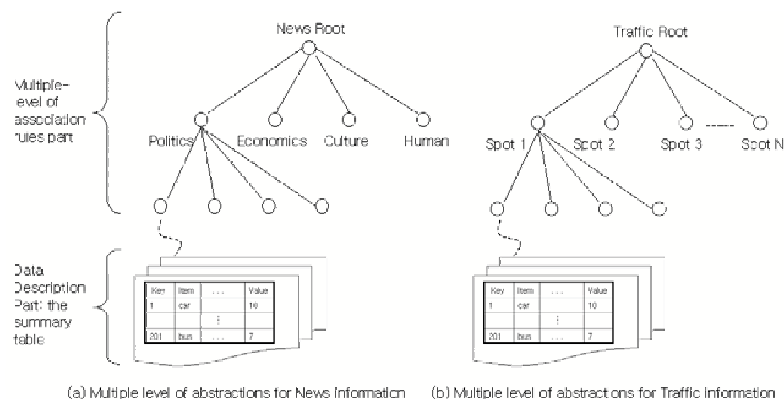
**Table 3.** Test sample data

Object No.	F1	F2	F3	F4	F5	D
1,2,3,4,5,6,7	0	X	X	X	X	0
1	1	1	Time	24%	2	1
2	1	1	Time	10%	3	2
3	1	0	X	4%	5	3
4	1	0	X	10%	4	4
5	1	0	X	50%	1	5
6	1	0	X	1%	7	6
7	1	0	X	1%	6	7

### 3.4 The association of the Web

An embedded hardware system has not enough memory devices to manage the data because it has a resource limitation problem and low performance capability. Actually, our system has a 512Mbyte working memory (NOR flash memory) and a 256Mbyte Compact Flash (CF) memory. The working memory includes operating system and some files to boot. It cannot store some information permanently. The CF memory includes 200Mbyte map data for car navigation, and 30Mbyte TTS DB. This is due to the cost of car navigation product. However, the user wants to utilize various information and services from a lot of different information sources. To resolve this problem, this system stores and manages only multiple level-of-abstraction. This mined data for multiple-level association rules are performed on the server-side. The data mining server plays a role in performing the Web mining. Mining typical user profiles and URL associations from the vast amount of access logs is an important feature. It deals with tailoring the interaction with Web information space based on information about the users.

The multiple level-of-abstraction is composed of multiple-level association rules and a summary table. The methods for mining associations at a generalized abstraction level by extension of the Apriori algorithm is applied as in [13]. The summary table forms the topic based indexing scheme. It stores basic information about groups of tuples of the underlying relations. This summary table is incrementally updateable and is able to support a variety of data mining and statistical analysis tasks. The summary table forming the indexed file is downloaded from the data mining server when the system is first switched on at the start of the day and the information is changed in the data mining server. The generalization process using attribute-oriented induction approach [14] for summary tables is performed on the server-side. It extracts a large set of relevant data in a database from a low concept level to a relatively high one. Thus, the system does not spend extra calculation time for data mining on an embedded system.



**Fig 4.** Multiple level-of-abstraction to manage the news and traffic information.

The sample structure of the multiple level-of-abstraction is as shown in Figure 4. It has a hierarchy form to index the data. We use two kinds of mined data; news and traffic information. (a) of Figure 4 depicts the news information. (b) of Figure 4 depicts the traffic information. The summary table basically includes the primary key, data, title, associated URL, and comments. Embedded applications that represent the news and traffic information just display the multiple level-of-abstraction information. If the user wants to see the specific information, that information is downloaded and displayed on the screen by selecting the specific button, or speaking the title. The speech interactive agent requests the URL for information to be sent to the data mining server, then the server sends the requested information in a form of HTML type text using HTTP protocol. The received text information is parsed and passed to the TTS, then the TTS reads this texts.

## 4. Discussions and Conclusions

### 4.1. Discussions

As the quality of automatic speech recognition (ASR) and text-to-speech (TTS) steadily improves, a variety of multimedia application services using embedded ASR (eASR), distributed speech recognition (DSR) and embedded TTS (eTTS) are being introduced for commercial use. In particular, since the demand of Telematics services is surging, speech interface to interact with human users has become an essential means of the multi-modal interface. As a Telematics client service interface, the eASR, DSR, and eTTS combined as a stand-alone unit provides an easy manipulation interface for command and controlling a car navigation system while the driver can pay attention to safe driving. In addition, as computer technology is improved, small sized computers such as AutoPCs has been utilized in various fields. Thus, by using this embedded system, the user requests and wants to utilize various service applications that they is used on a desktop PC, even while driving a car.

However, an embedded hardware system has a resource limitation and low performance capability. Actually, this condition is not able to represent huge data. Thus, a new architectural model is required in an embedded system. One alternative method is to use the Web. On the server-side, a comprehensive database is first mined, and then all the discovered patterns are stored in a DBMS. On the client-side, some abstraction data and indexes are stored. If the user wants to show specific data, the client obtains that information from a data mining server via the Internet using abstraction data and indexes. In our system, we reduce the memory size by using this concept. Even if a data communication fee per a packet should be paid, compression techniques for transmission packets could reduce the packet size. In addition, this can be resolved according to the policy of service usage.

With the above concepts, we designed a framework for command, control, communication, and intelligence environment based on a software agent on an embedded car navigation system, and then implemented it on AutoPC environment as shown in

Figure 5. The proposed framework provides the structure to extend the system easily and integrate with other services. It is possible that the core processing such as combining rules from interface modalities, data fusion rules, DBMS processing, and communication tasks are done by the speech interactive agent. In this system, a conversational tool provides advantages in confirming the final decision to use human interactive data mining [5].



**Fig 5.** Embedded system using an AutoPC for car navigation

#### 4.2. Conclusions

In this paper, we proposed a decision theoretic fusion framework that includes the multiple level-of-abstraction approach combining multiple-level association rules and the summary table, as well as the active interaction rule generation algorithm using the rough set theory for actionability on an embedded car navigation system. In addition, it included the sensory and data fusion level rule extraction algorithm to cope with simultaneous events occurring from multi-modal interface. Using such a decision theoretic fusion framework, a variety of applications can be applied easily to this system in the form of flexible, extensible and transparent ones. We expect that this fusion framework will be able to meet the user's demands and desires.

#### 5. Acknowledgements

This work was supported by grant No. 2003-218 from the Korea Institute of Industrial Technology Evaluation & Planning Foundation.

#### 6. References

- [1] Andrew Kusiak, and et al., "Autonomous Decision-Making: A Data Mining Approach," IEEE Trans. on Information Technology in Biomedicine, Vol. 4, No. 4, December 2000.

- [2] Sushmita Mitra, and et al., "Data Mining in Soft Computing Framework: A Survey," IEEE Trans. on Neural Networks, Vol. 13, No. 13, January 2002.
- [3] Ming-Syan Chen, and et al., "Data Mining: An Overview from a Database Perspective," IEEE Trans. on knowledge and Data Engineering, Vol. 8, No. 6, December 1996.
- [4] Yuval Elovici and Dan Braha, "A Decision-Theoretic Approach to Data Mining," IEEE Trans on Systems, Man, and Cybernetics – PART A:Systems and Humans, Vol. 33, No. 1, January 2003.
- [5] Charu C. Aggarwal, "A Human-Computer Interactive Method for Projected Clustering," IEEE Trans. On Knowledge and Data Engineering, Vol. 16, No. 4, April 2004.
- [6] Jiawei Han and et al. "Data mining for Web intelligence," Computer , Volume: 35 , Issue: 11 , Nov. 2002
- [7] Ashida, H. and Morita, T., "Architecture of data mining server: DATAFRONT/Server," IEEE SMC '99 Conference Proceedings., Volume: 5 , 12-15 Oct. 1999
- [8] X. Huang, A. Acero and H. Hon, ***Spoken Language Processing***, Prentice Hall PTR, 2001.
- [9] Takata, S.; Kawato, S.; Mase, K., "Conversational agent who achieves tasks while interacting with humans based on scenarios," Robot and Human Interactive Communication Proceedings:11th IEEE International Workshop, 25-27 Sept. 2002.
- [10] Aakay, M., Marsic, I., Medl, A., Guangming Bu, "A system for medical consultation and education using multimodal human/machine communication," IEEE Trans-Information Technology in Biomedicine, Vol 2 , Issue: 4 , Dec. 1998.
- [11] Richard T. Antony, ***Principles of Data Fusion Automation***, Artech house, 1995.
- [12] Heungkyu Lee, Ohil Kwon and Hanseok Ko, "Speech Interactive Agent System for Car Navigation Using Embedded ASR/TTS and DSR," 8th IEEE International Symposium on Consumer Electronics 2004.
- [13] R. Srikant and R. Agrawal, "Mining Generalized Association Rules," Proc. 21th Int'l Conf. Very Large Data Bases, pp. 407-419, Sept. 1995.
- [14] J. Han, Y. Cai, and N. Cercone, "Data-Driven Discovery of Quantitative Rules in Relational Databases," IEEE Trans. Knowledge and Data Eng., vol. 5, pp.29-40, 1993.

# Mining MOUCLAS Patterns and Jumping MOUCLAS Patterns to Construct Classifiers

Yalei Hao\*      Gerald Quirchmayr\*\* \*      Markus Stumptner\*

\*Advanced Computing Research Centre, University of South Australia, SA5095, Australia  
Yalei.Hao@postgrads.unisa.edu.au, Gerald.Quirchmayr@unisa.edu.au, mst@cs.unisa.edu.au

\*\*Institut für Informatik und Wirtschaftsinformatik, Universität Wien, Liebiggasse 4, A-1010 Wien, Austria

**Abstract** . This paper proposes a mining novel approach which consists of two new data mining algorithms for the classification over quantitative data, based on two new pattern called *MOUCLAS* (MOUntain function based CLASsification) Patterns and *Jumping MOUCLAS* Patterns. The motivation of the study is to develop two classifiers for quantitative attributes by the concepts of the association rule and the clustering. An illustration of using petroleum well logging data for oil/gas formation identification is presented in the paper. *MPs* and *JMPs* are ideally suitable to derive the implicit relationship between measured values (well logging data) and properties to be predicted (oil/gas formation or not). As a hybrid of classification and clustering and association rules mining, our approach have several advantages which are (1) it has a solid mathematical foundation and compact mathematical description of classifiers, (2) it does not require discretization, (3) it is robust when handling noisy or incomplete data in high dimensional data space.

## 1 Introduction

Data mining based classification aims to build accurate and efficient classifiers not only on small data sets but more importantly also on large and high dimensional data sets, while the widely used traditional statistical data analysis techniques are not sufficiently powerful for this task<sup>1,2</sup>. With the development of new data mining techniques on association rules, new classification approaches based on concepts from association rule mining are emerging. These include such classifiers as ARCS<sup>3</sup>, CBA<sup>4</sup>, LB<sup>5</sup>, JEP<sup>6</sup>, etc., which are different from the classic decision tree based classifier C4.5<sup>7</sup> and k-nearest neighbor<sup>8</sup> in both the learning and testing phases. To improve ARCS<sup>3</sup>, A non-grid-based technique<sup>9</sup> has been further proposed to find quantitative association rules that can have more than two predicates in the antecedent. All the above algorithms are constrained by the framework of binning. Though several excellent discretization algorithms<sup>10,11</sup> are proposed, a standard approach to discretization has not yet been developed.

Therefore, all the above research issues establish a challenge, which is whether it is possible that an association rule based classifier with any number of predicates in the antecedent can be developed for quantitative attributes by the concepts of clustering which can overcome the limitation caused by the discretization method. In this paper, to resolve the problem, we present a new approach to the classification over quantitative data in high dimensional databases, called *MOUCLAS* (MOUntain function based CLASsification), based on the concept of the fuzzy set membership function. It aims at integrating the advantages of classification, clustering and association rules mining to identify interesting patterns in selected sample data sets.

## 2 Problem Statement

We now give a formal statement of the problem of *MOUCLAS* Patterns (called *MPs*) and introduce some definitions.

The *MOUCLAS* algorithm, similar to ARCS, assumes that the initial association rules can be agglomerated into clustering regions, while obeying the anti-monotone rule constraint. Our proposed framework assumes that the training dataset  $D$  is a normal relational set, where transaction  $d \in D$ . Each transaction  $d$  is described by attributes  $A_j$ ,  $j = 1$  to  $l$ . The dimension of  $D$  is  $l$ , the number of attributes used in  $D$ . This allows us to describe a database in terms of volume and dimension.  $D$  can be classified into a set of known classes  $Y$ ,  $y \in Y$ . The value of an attribute must be quantitative. In this work, we treat all the attributes uniformly. We can treat a transaction as a set of (attributes, value) pairs and a class label. We call each (attribute, value) pair an item. A set of items is simply called an itemset.

In this paper, we propose two novel classifiers, called the *De-MP* and *J-MP*, which exploit the discrimination ability of *MOUCLAS* Patterns (*MPs*) and *Jumping MOUCLAS* Patterns (*JMPs*).

The *MOUCLAS* Pattern (so called *MP*) has an implication of the form:

$$Cluster(D)_i \rightarrow y,$$

where  $Cluster(D)_i$  is a cluster of  $D$ ,  $i = 1$  to  $m$ , and  $y$  is a class label.

The definitions of *frequency* and *accuracy* of MOUCLAS Patterns are defined as following: The *MP* satisfying minimum support is **frequent**, where *MP* has support  $s$  if  $s\%$  of the transactions in  $D$  belong to  $Cluster(D)_i$  and are labeled with class  $y$ . The *MP* that satisfies a pre-specified minimum confidence is called **accurate**, where *MP* has confidence  $c$  if  $c\%$  of the transactions belonging to  $Cluster(D)_i$  are labeled with class  $y$ .

We also adopt the concept of reliability<sup>12</sup> to describe the correlation. The measure of reliability of the association rule  $A \Rightarrow B$  can be defined as:

$$\text{reliability } R(A \Rightarrow B) = \left| \frac{P(A \wedge B)}{P(A)} - P(B) \right|$$

Since  $R$  is the difference between the conditional probability of  $B$  given  $A$  and the unconditional of  $B$ , it measures the effect of available information of  $A$  on the probability of the association rule. Correspondingly, the greater  $R$  is, the stronger MOUCLAS patterns are, which means the occurrence of  $Cluster(D)_i$  more strongly implies the occurrence of  $y$ . Therefore, we can utilize reliability to further prune the selected *frequent and accurate and reliable* MOUCLAS patterns (*MPs*) to identify the truly interesting *MPs* and make the discovered *MPs* more understandable. The *MP* satisfying minimum reliability is **reliable**, where *MP* has reliability defined by the above formula.

Given a set of transactions,  $D$ , the problems of *De-MP* are to discover *MPs* that have support and confidence greater than the user-specified minimum support threshold (called *minsup*)<sup>13</sup>, and minimum confidence threshold (called *minconf*)<sup>13</sup> and minimum reliability threshold (called *minR*) respectively, and to construct a classifier based upon *MPs*.

A Jumping MOUCLAS Pattern (*JMP*) can be further defined based on the notion of the Jumping Emerging Pattern<sup>6</sup> (*JEP*) and *MP*. A *JEP* is an itemset whose support increases significantly from 0 in a class (say poisonous class in mushroom data from the UCI repository) to a user-specified value in another class (say edible class). We can then use *JEP* as an index for dimensionality reduction. For each *JEP* in a certain class  $y$ , only the attributes of the *JEP* will be kept for all the transactions in the class  $y$ . We then perform the clustering on those transactions.

Let  $C$  denote the dataset of transaction  $d$  labeled with class  $y$  after dimensionality reduction processing by *JEPs*. A *JMP* can be defined as a *cluster\_rule*, namely a rule:

$$cluset \rightarrow y,$$

where *cluset* is a set of itemsets from a cluster  $Cluster(C)_i$ , which is obtained from the clustering on the same class of transactions after dimensionality reduction via *JEP*,  $y$  is a class label,  $y \in Y$ . Let *JMPset* denote a set of *JMPs* which corresponds to the same *JEP*.

Suppose the number of transactions of  $C$  in *cluset* is *cluCount*, the number of transactions in  $C$  is *clasCount*, the *support* of transaction  $d$  belong to *cluset* in  $C$ , denoted as *subsup*, can be defined by the formula:

$$\text{subsup} = \frac{\text{cluCount}}{\text{clasCount}}$$

Given a set of transactions,  $D$ , the problems of *J-MP* is to discover all *JMPs* and calculate their *subsup* and construct a classifier based upon *JMPs*.

### 3 The MOUCLAS-1 Algorithm

The classification technique, MOUCLAS-1, consists of two steps:

1. Discovery of *frequent, accurate and reliable* *MPs*.
2. Construction of a classifier, called *De-MP*, based on *MPs*.

The core of the first step in the MOUCLAS-1 algorithm is to find all *cluster\_rules* that have support above *minsup*. Let  $C$  denote the dataset  $D$  after dimensionality reduction processing. A *cluster\_rule* represents a *MP*, namely a rule:

$$cluset \rightarrow y,$$

where *cluset* is a set of itemsets from a cluster  $Cluster(C)_i$ ,  $y$  is a class label,  $y \in Y$ . The support count of the *cluset* (called *clusupCount*) is the number of transactions in  $C$  that belong to the *cluset*. The support count of the *cluster\_rule* (called *cisupCount*) is the number of transactions in  $D$  that belong to the *cluset* and are labeled with class  $y$ . The *confidence* of a *cluster\_rule* is  $(\text{cisupCount} / \text{clusupCount}) \times 100\%$ . The support count of the class  $y$  (called *clasupCount*) is the number of transactions in  $C$  that belong to the class  $y$ . The *support* of a class (called *clasup*) is  $(\text{clasupCount} / |C|) \times 100\%$ , where  $|C|$  is the size of the dataset  $C$ .

Given a *MP*, the *reliability*  $R$  can be defined as:

$$R(\text{cluset} \rightarrow y) = \left| (\text{cisupCount} / \text{clusupCount}) - (\text{clasupCount} / |C|) \right| \times 100\%$$

The traditional association rule mining only uses a single *minsup* in rule generation, which is inadequate for many practical datasets with uneven class frequency distributions. As a result, it may happen that the rules found for infrequent classes are insufficient and too many may be found for frequent classes, inducing

useless or over-fitting rules, if the single *minsup* value is too high or too low. To overcome this drawback, we apply the theory of mining with multiple minimum supports<sup>14</sup> in the step of discovering the frequent MPs as following.

Suppose the total support is *t-minsup*, the different minimum class support for each class *y*, denoted as *minsup<sub>i</sub>*, can be defined by the formula:

$$\text{minsup}_i = t\text{-minsup} \times \text{freqDistr}(y)$$

where, *freqDistr(y)* is the function of class distributions. *Cluster\_rules* that satisfy *minsup<sub>i</sub>*, are called *frequent cluster\_rules*, while the rest are called *infrequent cluster\_rules*. If the *confidence* is greater than *minconf*, we say the *MP* is *accurate*.

The first step of *MOUCLAS-1* algorithm works in three sub-steps, by which the problem of discovering a set of *MPs* is solved:

**Algorithm:** Mining *frequent* and *accurate* and *reliable* *MOUCLAS* patterns (*MPs*)

**Input:** A training transaction database, *D*; minimum support threshold (*minsup<sub>i</sub>*); minimum confidence threshold (*minconf*); minimum reliability threshold (*minR*)

**Output:** A set of *frequent*, *accurate* and *reliable* *MOUCLAS* patterns (*MPs*)

**Methods:**

- (1) Reduce the dimensionality of transactions *d*, which efficiently reduces the data size by removing irrelevant or redundant attributes (or dimensions) from the training data, and
- (2) Identify the clusters of database *C* for all transactions *d* after dimensionality reduction on attributes *A<sub>j</sub>* in database *C*, based on the Mountain function, which is a fuzzy set membership function, and specially capable of transforming quantitative values of attributes in transactions into linguistic terms, and
- (3) Generate a set of *MPs* that are both *frequent*, *accurate* and *reliable*, namely, which satisfy the user-specified minimum support (called *minsup<sub>i</sub>*), minimum confidence (called *minconf*) and minimum reliability (called *minR*) constraints.

In the first sub-step, we reduce the dimensionality of transactions in order to enhance the quality of data mining and decrease the computational cost of the *MOUCLAS* algorithm. Since, for attributes *A<sub>j</sub>*, *j* = 1 to *l* in database, *D*, an exhaustive search for the optimal subset of attributes within  $2^l$  possible subsets can be prohibitively expensive, especially in high dimensional databases, we use heuristic methods to reduce the search space. Such greedy methods are effective in practice, and include such techniques as stepwise forward selection, stepwise backward elimination, combination of forwards selection and backward elimination, etc. The first sub-step is particularly important when dealing with raw data sets. Detailed methods concerning dimensionality reduction can be found in some papers<sup>15-18</sup>.

Fuzzy based clustering is performed in the second sub-step to find the clusters of quantitative data. The Mountain-climb technique proposed by R. R. Yager and D. P. Filev<sup>19</sup> employed the concept of a mountain function, a fuzzy set membership function, in determining cluster centers used to initialize a Neuro-Fuzzy system. The subtractive clustering technique<sup>20</sup> was defined as an improvement of Mountain-climb clustering. A similar approach is provided by the DENCLUE algorithm<sup>21</sup>, which is especially efficient for clustering on high dimensional databases with noise. The techniques of Mountain-climb clustering, Subtractive clustering and Denclue provide an effective way of dealing with quantitative attributes by mountain functions (or influence functions), which has a solid mathematical foundation and compact mathematical description and is totally different from the traditional processing method of binning. It offers us an opportunity of mining the patterns of data from an innovative angle. As a result, part of the research task presented in the introduction can now be favorably answered.

The observation that, a region which is dense in a particular subspace must create dense regions when projected onto lower dimensional subspaces, has been proved by R. Agrawal and his research cooperators in CLIQUE<sup>22</sup>. In other words, the observation follows the concepts of the apriori property. Hence, we may employ prior knowledge of items in the search space based on the property so that portions of the space can be pruned. The successful performance of CLIQUE has again proved the feasibility of applying the concept of apriori property to clustering. It brings us a step further towards the solution of the rest part of the research task, that is, if the initial association rules can be agglomerated into clustering regions, just like the condition in ARCS, we may be able to design a new classifier for the purpose of classification, which confines its search for the classifier to the cluster of dense units of high dimensional space. The answer to the rest research task can contribute to the third sub-step of the *MOUCLAS* algorithm to the forming of the antecedent of *cluster\_rules*, with any number of predicates in the antecedent. In the third sub-step, we identify the candidate *cluster\_rules* which are actually *frequent* and *accurate* and *reliable*. From this set of *frequent* and *accurate* and *reliable cluster\_rules*, we produce a set of *MPs*.

Let *I* be the set of all items in *D*, *C* be the dataset *D* after dimensionality reduction, where transaction *d* ∈ *C* contains  $X \subseteq I$ , a *k*-itemset. Let *E* denote the set of candidates of *cluster\_rules*, where *e* ∈ *E*, and *F* denote the set of frequent *cluster\_rules*. The first step of the *MOUCLAS* algorithm is given in Figure 1 as follows.

1 *X* = reduceDim (*I*); // reduce the dimensionality on the set of all items *I* of in *D*

2 *Cluster(C)<sub>i</sub>* = genCluster (*C*); // identify the complete clusters of *C*

3 **for** each *Cluster(C)<sub>i</sub>*, **do**

*E* = genClusterrules(*cluset*, *class*); // generate a set of candidate *cluster\_rules*

```

4  for each transaction  $d \in C$  do
5       $E_d = \text{genSubClusterrules}(E, d)$ ; // find all the cluster_rules in  $E$  whose cluset are supported by  $d$ 
6      for each  $e \in E_d$  do
7           $e.\text{clusupCount}++$ ; // accumulate the clusupCount of the cluset of cluster_rule  $e$ 
8          if  $d.\text{class} = e.\text{class}$  then  $e.\text{cisupCount}++$  // accumulate the cisupCount of cluster_rule  $e$ 
                                     supported by  $d$ 
9      end
10 end
11  $F = \{e \in E \mid e.\text{cisupCount} \geq \text{minsup}_i\}$ ; // construct the set of frequent cluster_rules
12  $MP = \text{genRules}(F)$ ; //generate  $MP$  using the genRules function by minconf and minR
13 end
14  $MPs = \cup MP$ ; // discover the final set of  $MPs$ 
    
```

Figure 1: The First Step of the MOUCLAS-1 Algorithm

The task of the second step in *MOUCLAS-1* algorithm is to use a heuristic method to generate a classifier, named *De-MP*, where the discovered  $MPs$  can cover  $D$  and are organized according to a decreasing precedence based on their confidence and support. Suppose  $R$  be the set of *frequent, accurate and reliable*  $MPs$  which are generated in the past step, and  $MP_{\text{default\_class}}$  denotes the default class, which has the lowest precedence. We can then present the *De-MP* classifier in the form of  $\langle MP_1, MP_2, \dots, MP_n, MP_{\text{default\_class}} \rangle$ ,

where  $MP_i \in R$ ,  $i = 1$  to  $n$ ,  $MP_a > MP_b$  if  $n \geq b > a \geq 1$  and  $a, b \in i$ ,  $C \subseteq \cup \text{cluset of } MP_i$ .

The second step of the *MOUCLAS-1* algorithm also consists of three sub-steps, by which the *De-MP* classifier is formed:

**Algorithm:** Constructing *De-MP* Classifier

**Input:** A training database after dimensionality reduction,  $C$ ; The set of *frequent and accurate and reliable* *MOUCLAS* patterns ( $MPs$ )

**Output:** *De-MP* Classifier

**Methods:**

(1) Identify the order of all discovered  $MPs$  based on the definition of precedence and sequence them according to decreasing precedence order.

(2) Determine possible  $MPs$  for *De-MP* classifier from  $R$  following the descending sequence of  $MPs$ .

(3) Discard the  $MPs$  which cannot contribute to the improvement of the accuracy of the *De-MP* classifier and keep the final set of  $MPs$  to construct the *De-MP* classifier.

In the first sub-step, the  $MPs$  are sorted in descending order, which has the training transactions surely covered by the  $MPs$  with the highest precedence when possible in the next sub-step. The sort of the whole set of  $MPs$  is performed following the definition of *precedence*:

Given two  $MPs$ , we say that  $MP_a$  has a higher precedence than  $MP_b$ , denoted as  $MP_a > MP_b$ ,

if  $\forall MP_a, MP_b \in MPs$ , it holds that: the confidence of  $MP_a$  is greater than that of  $MP_b$ , or if their confidences are the same, but the support of  $MP_a$  is greater than that of  $MP_b$ , or if both the confidences and supports of  $MP_a$  and  $MP_b$  are the same, but  $MP_a$  is generated earlier than  $MP_b$ .

In the second sub-step, we test the  $MPs$  following decreasing precedence and stop the sub-step when there is no rule or no training transaction. For each  $MP$ , we scan  $C$  to find those transactions satisfying the *cluset* of the  $MP$ . If the  $MP$  can correctly classify one transaction, we store it in a set denoted as  $L$ . Those transactions satisfying the *cluset* of the  $MP$  will be removed from  $C$  at each pass. Each transaction can be identified by a unique ID. The next pass will be performed on the remaining data. A default class is defined at each scan, which is the majority class in the remaining data. At the end of each pass, the total number of errors that are made by the current  $L$  and the default class are also stored. When there is no rule or no training transaction left, we terminate this sub-step. After this sub-step, every  $MP$  in  $L$  can correctly classify at least one training transaction in  $C$ .

In the third sub-step, though we would like to find as many  $MPs$  as possible to give good coverage of the training transactions in the second sub-step, we prefer strong  $MPs$  which have relatively high support and confidence, due to their characteristics of corresponding to larger coverage and stronger differentiating power. Meanwhile, we hope that the *De-MP* classifier, consisting of a combination of strong  $MPs$ , has a relatively smaller number of classification errors, because of greedy strategy. In addition, the reduction of  $MPs$  can increase the understandability of the classifier. Therefore, in this sub-step, we identify the first  $MP$  with the least number of errors in  $L$  and discard all the  $MPs$  after it because these  $MPs$  produce more errors. The undiscarded  $MPs$  and the default class corresponding to the first  $MP$  with the least number of errors in  $L$  form our *De-MP* classifier.

The second step of the *MOUCLAS* algorithm is shown in Figure 2.

```

1  $R = \text{sort}(R)$ ; // sort  $MPs$  based on their precedence
    
```

```

2 for each  $MP \in R$  in sequence do
3    $temp = \emptyset$  ;
4   for each transaction  $d \in C$  do
5     if  $d$  satisfies the cluset of  $MP$  then
6       store  $d.ID$  in  $temp$ ;
7       if  $MP$  correctly classifies  $d$  then
8         insert  $MP$  at the end of  $L$ ;
9       delete the transaction who has ID in  $temp$  from  $C$ ;
10      selecting a default class for the current  $L$ ; // determine the default class based on majority class of
                                                remaining transactions in  $C$ 
11    end
12    compute the total number of errors of  $L$ ; // compute the total number of errors that are made by the
                                                current  $L$  and the default class
13  end
14 Find the first  $MP$  in  $L$  with the lowest total number of errors and discard all the  $MPs$  after the  $MP$  in  $L$ ;
15 Add the default class associated with the above mentioned first  $MP$  to end of  $L$ ;
16  $De-MP$  classifier =  $L$ 
    
```

Figure 2: The Second Step of the MOUCLAS Algorithm

In the testing phase, when we classify a new transaction, the first  $MP$  in  $De-MP$  satisfying the transaction is used to classify it. In  $De-MP$  classifier, *default\_class*, having the lowest precedence, is used to specify a default class for any new sample that is not satisfied by any other  $MPs$  as in C4.5<sup>7</sup>, CBA<sup>4</sup>.

#### 4 The MOUCLAS-2 Algorithm

The classification technique, *MOUCLAS-2*, consists of two main processes:

1. Discovering of all *JMPs* for each class.
2. Calculating their *subsup* and building a classifier, called *J-MP*, based on *JMPs*.

The core of the *MOUCLAS-2* algorithm is to find all *cluster\_rules*, namely the *JMPs*. The *MOUCLAS-2* algorithm works in three sub-steps, by which the problem of discovering *JMPsets* and construction of a classifier is solved:

**Algorithm:** Mining Jumping *MOUCLAS* Patterns (*JMPs*) and building *J-MP* Classifier

**Input:** A training transaction database,  $D$ ;

**Output:** *J-MP* Classifier

**Methods:**

- (1) Reduce the dimensionality of transactions  $d$  in each class  $y$  by the information of the attributes in corresponding *JEPs*, and
- (2) Identify all the clusters of database based on the Mountain function, which is a fuzzy set membership function, and specially capable of transforming quantitative values of attributes in transactions into linguistic terms, and
- (3) Generate *JMPsets* for each class  $y$  and calculate their *subsup*.

In the first sub-step, detailed method concerning JEP can be found in this paper<sup>6</sup>.

The third sub-step of the *MOUCLAS-2* algorithm form the *cluster\_rules*, with any number of predicates in the antecedent. It brings us a step further towards the solution of our research challenge. From this set of *cluster\_rules* of a class  $y$ , we produce a set of *JMPs* for the class  $y$ .

Let  $\bar{I}$  be the set of all items in  $D$  labeled with class  $y$ ,  $C$  be the dataset of transaction  $d$  labeled with class  $y$  after dimensionality reduction processing by a *JEP*, where transaction  $d \in C$  contains  $X_i \subseteq \bar{I}$ , a  $k$ -itemset, and  $i$  be the number of *JEPs* in the class  $y$ . Let  $E$  denote a set of *cluster\_rules* (*JMPset*) of a class  $y$ , corresponding to a JEP, where  $e \in E$ .

The first step of the *MOUCLAS-2* algorithm is given in Figure 3 as follows.

```

1  $X = \text{genJEP}(I)$ ; // generate all the JEPs of all the class  $y$  in  $D$ 
2 for each class  $y$  do
3   for each JEP of a same class  $y$  do
4      $X_i = \text{reduceDim}(I)$ ; // reduce the dimensionality on the set of all items  $I$  in  $D$  labeled with class  $y$  based
    on the attributes of the JEP
5      $E_i = \text{genClusterrules}(\text{cluset}, \text{class})$ ; // generate a set of cluster_rules, namely JMPset, based on  $X_i$ 
6   for each transaction  $d \in C$  do
    
```

```

7   if one  $e \in E_i$  can be supported by  $d$  then  $e.cluCount++$ ; // accumulate the  $cluCount$  of  $cluster\_rule$   $e$ 
   supported by  $d$ 
8   end
9    $subsup_i = \frac{e.cluCount}{|C|}$ ; // calculate the  $subsup$  of each  $JMPset$ 

10  end
11  end
12   $JMPs = \cup E_i$ ; // discover the final set of  $JMP$ 
    
```

Figure 3: The Training Phase of the MOUCLAS-2 Algorithm

In the testing phase, The MOUCLAS-2 algorithm also consists of two sub-steps, by which the  $J$ -MP classifier can classify test data:

**Algorithm:** Classification Process of  $J$ -MP Classifier

**Input:** A test database,  $D$ ; The set of Jumping MOUCLAS patterns ( $JMPs$ ); The *support* of transaction  $d$  belong to  $JMPs$  in  $C$  ( $subsup$ )

**Output:** classification result of test database

**Methods:**

- (1) Determine the  $subsup$  of each transaction  $d$  in  $D$  in each class.
- (2) Classify the test data.

In the first sub-step, we firstly determine whether a  $JMPset$  can be supported by a transaction  $d \in D$ . If so, we then sum up the total  $subsup$  of the transaction  $d$  in one class. In this way, the  $subsup_y$  of the transaction  $d$  in the class  $y$  can be obtained, where  $y \in Y$ . In the second step, the testing transaction  $d$  can be labeled as the class  $y$ , where the  $subsup_y$  is greater than all the others. If the transaction  $d$  has the same maximum  $subsup_y$  in two classes, then it is labeled as the class, whose  $JMPs$  are generated earlier than the other.

The classification process of the MOUCLAS algorithm is shown in Figure 4.

```

1  for each transaction  $d \in D$  do
2    for each class  $y$  do
3      for each  $JMPset$  of a same class  $y$  do
4        if  $d$  satisfies a  $JMPset$  then  $e.subsupt++$ ; // accumulate the  $subsup$  of  $JMPsets$  supported by  $d$ 
5      end
6      the  $subsup_y$  of  $d$  in class  $y = e.subsupt$ ; // calculate the total  $subsup$  of  $d$  in class  $y$ 
7    end
8    if  $subsup_y$  is the maximum then  $d$  is labeled as  $y$ 
9    if the  $subsup$  in two or more classes are the same then  $d$  is labeled as the class, whose  $JMPs$  are
       generated earlier than the others.
10   if the  $subsup = 0$  then  $d$  is labeled as a default class
11 end
    
```

Figure 4: The Testing Phase of the MOUCLAS Algorithm

## 5 Example of MOUCLAS Application in Reservoir Characterization

Oil/gas formation identification is a vital task of reservoir characterization in the petroleum industry, where the petroleum database contains such records (or attributes) as seismic data, various types of well logging data and petrophysical property data whose values are all quantitative.

An illustration of using well logging data for purpose of oil/gas formation identification is illustrated in Figure 5. The well logging data sets include attributes (well logging curves) of GR (gamma ray), RDEV (deep resistivity), RMEV (shallow resistivity), RXO (flushed zone resistivity), RHOB (bulk density), NPHI (neutron porosity), PEF (photoelectric factor) and DT (sonic travel time). Since most of the reservoirs are horizontally and vertically heterogeneous, no depth information is used for training.

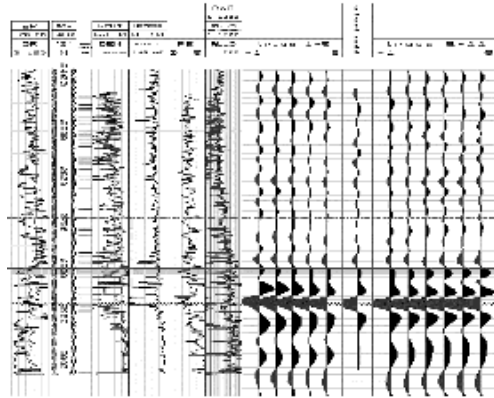


Figure 5: Quantitative Petroleum Data for MOUCLAS Mining  
(note: the dashed indicate the location of oil formation)

One transaction of the database can be treated as a set of the items corresponding to the same depth and a class label (oil/gas formation or not). A hypothetically useful *MP* or *JMP* may suggest a relation between well logging data and the class label of oil/gas formation since. In this sense, a selected set of such *MPs* or *JMPs* can be a useful guide to petroleum engineers to identify possible drilling targets and their depth and thickness at the stage of exploration and exploitation.

*MPs* and *JMPs* aim at deriving an explicit or implicit heuristic relationship between measured values (well logging data) and properties to be predicted (oil/gas formation or not). The *MOUCLAS* based method is ideally suitable to establish such implicit relationships through proper training. The notable advantage of *MOUCLAS* based algorithms over more traditional processing techniques such as model based well logging analysis is that a physical model to describe the relationship between the well logging data and the property of interest is not needed; nor is an very precise understanding of the physical phenomena of the well logging data. From this point of view, *MOUCLAS* based algorithms provides a complementary and useful technical approach towards the interpretation of petroleum data and benefits petroleum discovery.

## 6 Conclusions

Two novel classification patterns, the *MOUCLAS* Pattern (*MP*) and the *Jumping MOUCLAS* Pattern (*JMP*) for quantitative data in high dimensional databases, are investigated in this paper. We also propose the algorithm for the discovery of the interesting *MPs* and *JMPs* and construct two new classifiers called *De-MP* and *J-MP*. As a hybrid of classification and clustering and association rules mining, our approach may have several advantages which are (1) it has a solid mathematical foundation and compact mathematical description of classifiers, (2) it does not require discretization, as opposed to other, otherwise quite similar methods such as ARCS are strongly related to, (3) it is robust when handling noisy or incomplete data in high dimensional data space, regardless of the database size, due to its grid-based characteristic. An illustration of application of *MPs* and *JMPs* is presented for the cost effective and intelligent well logging data analysis for reservoir characterization. In the future research, we attempt to carry out experiments on petroleum datasets to establish a relationship between different well logs, seismic attributes, laboratory measurements and other reservoir properties to evaluate performance of the *MOUCLAS* algorithms proposed in this paper.

## 7 Acknowledgement

This work was partially supported by the Australia-China Special Fund for Scientific and Technological Cooperation under grant CH030086.

## References

1. Fayyad, U. M., Piatetsky-Shapiro, G., & Smyth, P. From data mining to knowledge discovery: An overview. *Advances in knowledge discovery and data mining*. AAAI/MIT Press. (1996) 1-34
2. Han, J., & M. Kamber. *Data mining: concepts and techniques*. Morgan Kaufmann Publishers. (2000)
3. B. Lent, A. Swami, and J. Widom. Clustering association rules. *ICDE'97*, (1997) 220-231

4. B. Liu, W.Hsu, and Y.Ma. Integrating classification and association rule mining. KDD'98. (1998) 80-86
5. Meretakis, D., & Wuthrich, B. Extending naive Bayes classifiers using long itemsets. Proc. of the Fifth ACM SIGKDD. ACM Press. (1999) 165-174
6. Jinyan Li, Guozhu Dong, Kotagiri Ramamohanarao. Making Use of the Most Expressive Jumping Emerging Patterns for Classification. Knowledge and Information Systems, 3(2):131--145, 2001.
7. Quinlan, J. R. C4.5: Programs for machine learning. San Mateo, CA: Morgan Kaufmann. (1993)
8. Cover, T. M., & Hart, P. E. Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13. (1967) 21-27
9. R. Skikant and R. Agrawal. Mining quantitative association rules in large relational tables. SIG-MOD'96, (1996) 1-12.
10. Fayyad, U., & Irani, K. Multi-interval discretization of continuous-valued attributes for classification learning. Proc. of the 13th Int'l Conf. on Artificial Intelligence. Morgan Kaufmann. (1993) 1022--1029
11. Dougherty, J., Kohavi, R., & Sahami, M. Supervised and unsupervised discretization of continuous features. Proc. of the Twelfth Int'l Conf. on Machine Learning pp. 94--202. Morgan Kaufmann. (1995)
12. Khalil M. Ahmed, Nagwa M. El-Makky, Yousry Taha: A note on "Beyond Market Baskets: Generalizing Association Rules to Correlations". In The Proceedings of SIGKDD Explorations Volume1, Issue 2, (2000) 46-48
13. Agrawal, R., Srikant, R. Fast algorithms for mining association rules. Proc. of the 20th VLDB (1994) 487- 499
14. Bing Liu, Wynne Hsu, Yiming Ma, "Mining Association Rules with Multiple Minimum Supports" Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD-99), August 15-18, San Diego, CA, USA (1999)
15. Dong, G., & Li, J. Feature selection methods for classification. Intelligent Data Analysis: An International Journal, 1, (1997)
16. H. Liu and H. Motoda, editors. Feature Selection for Knowledge Discovery and Data Mining. Boston: Kluwer Academic Publishers, (1998)
17. W.Sarawagi and M. Stonebraker. On automatic feature selection. Int'l J. of Pattern Recognition and Artificial Intelligence, 2, (1988) 197-220.
18. R. Kohavi and G. John. Wrappers for feature subset selection. Artificial Intelligence, (1997) 273-324
19. Yager, R. and D. Filev, "Generation of Fuzzy Rules by Mountain Clustering," Journal of Intelligent & Fuzzy Systems, Vol. 2, No. 3, (1994) 209-219
20. Chiu, S. L. Fuzzy model identification based on cluster estimation. Journal of Intelligent and Fuzzy System, 2(3), (1994)
21. A. Hinneburg and D. Keim. An efficient approach to clustering in large Multimedia dataset with noise. KDD'98, (1998) 58-65
22. R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. SIGMOD'98. (1998)

# Exploratory Health Data Mining: Identifying Factors Associated with Colorectal Cancer

Jie Chen<sup>1</sup>, Hongxing He<sup>1</sup>, Huidong Jin<sup>1</sup>, Damien McAullay<sup>1</sup>, Graham Williams<sup>1,2</sup>, and Chris Kelman<sup>3</sup>

<sup>1</sup>CSIRO Mathematical and Information Sciences  
GPO Box 664, Canberra ACT 2601, Australia  
`Firstname.Lastname@csiro.au`

<sup>2</sup>Current address: Australian Taxation Office  
`Graham.Williams@togaware.com`

<sup>3</sup>Department of Health and Ageing(DoHA)  
`Chris.Kelman@anu.edu.au`

**Abstract.** This paper explores data mining techniques for the task of identifying and describing factors which may affect the occurrence and prevalence of colorectal cancer (CRC) from population based administrative health data. Association rule discovery, association classification and scalable clustering analysis are applied to the colorectal cancer patients' profiles in contrast to background patients' profiles. These data mining methods enable us to identify the most common characteristics of the colorectal cancer patients. The knowledge discovered by data mining methods which are quite different from traditional survey approaches, although heuristic, may be useful in predicting risk factors leading to colorectal cancer.

## 1 Introduction

Colorectal cancer (cancer of the colon or rectum, abbreviated as CRC) is the second leading cause of cancer-related deaths in the United States for both men and women combined. The disease surpasses both breast and prostate cancer in mortality, and is second only to lung cancer in cause of cancer deaths. Despite the fact that it is highly preventable, approximately 146,940 new cases of colorectal cancer will be diagnosed in 2004 and more than 56,000 people will die from the disease in USA [1]. An almost equal number of men and women are diagnosed each year.

In Australia, colorectal cancer is the third most common cause of death from cancer in women (after breast and lung cancer) and men (after lung and prostate cancer). The exact cause of colorectal cancer is unknown, in fact it is thought that there is not one single cause. It is more likely that a number of factors, some known and many unknown, may work together to trigger the development of colorectal cancer.

Previous studies have identified risk factors which may increase a person's risk of developing colorectal cancer. However, having one or even several of these characteristics does not mean that a person is certain to develop the disease. The following factors are widely accepted risks:

- **Age.** Increasing age is considered a major risk factor for developing colorectal cancer. Colorectal cancer is rare in people under 40. The risk increases after the age of 40, rising sharply and progressively after the age of 50.
- **Dietary factors.** It is estimated that rates of colorectal cancer could be reduced in western populations by up to 35% through changes to the food we eat. A diet that is high in fat and low in fibre and vegetables has been linked with an increased risk of colorectal cancer. There has also been an association between heavily browned or charred meat and colorectal cancer. Excessive alcohol intake and a diet low in calcium have also been implicated.
- **Behavioural and lifestyle factors.** An inactive lifestyle, obesity and smoking have been associated with an increased risk of developing colorectal cancer.
- **Regional factors.** People in western countries such as Australia, America and New Zealand have a higher incidence of colorectal cancer than people in Asian or African countries. This may be partly due to differences in diet.

This paper aims at studying the relationship between CRC prevalence and various attributes of the patients. These attributes include demographics, medical service history etc. We use large administrative data sets instead of data from survey. The advantages are large data size, low cost and lack of selection bias. In our case, our dataset covers the medical records of more than one million people. The disadvantage is that we can not design what information we get from individual patients as in designed survey data. Since the transaction data is collected for administrative purposes only, some important information regarding patients' diet and lifestyle is missing. Therefore some well known risk factors can not be verified using administrative data. Nevertheless, it is a good practise to discover unexpected and interesting relationships from this dataset. Since the large data are to be explored, traditional methods dealing with small samples are not working well, we employ various data mining techniques in our analysis. Exploratory tools of data mining on large dataset may be able to identify some factors previously unnoticed.

The rest of the paper is organised as follows. Section 2 discusses related work. Section 3 describes the dataset and features selected for the mining process. Sections 4, 5 and 6 describe the methods and mined results for association rule discovery, scalable cluster analysis and association classification analysis respectively. Section 7 completes the paper with conclusion and discussion.

## 2 Related Work

Health data mining is a rewarding but highly challenging area [12, 3]. Recently, a few data mining and statistical analysis projects have been done for the surveillance and analysis of colorectal cancer patients. A Bayesian framework to extract

recurrence, the key outcome for measuring treatment effectiveness, for colorectal cancer patients, has been built in [11]. Logistic regression [10] and survival analysis [13] have been applied to identify recurrences and to model the prognosis of colorectal cancer patients. Different from these studies, this paper aims at identifying and describing factors which may affect the occurrence and prevalence of colorectal cancer in a new way. This paper applies various data mining techniques on linked administrative health dataset QLDS. In [2, 4], Adverse drug reaction has been successfully identified from the same dataset using association and classification algorithms.

### 3 Data

#### 3.1 QLDS

We use the Queensland Linked Data Set (QLDS) [15] for this exploratory data mining study. The QLDS has been made available under an agreement between Queensland Health and the Australian Department of Health and Ageing (DoHA). This data set links de-identified patient level hospital separation data (for the period between 1 July 1995 and 30 June 1999), Medicare Benefits Scheme (MBS) data, and Pharmaceutical Benefits Scheme (PBS) data (1 January 1995 to 31 December 1999) in Queensland.

Each record in the hospital data corresponds to one inpatient episode. Each record in the MBS corresponds to one MBS service for one patient. Each record in the PBS corresponds to one prescription service for one patient. As a result, each patient may have more than one hospital, MBS, and PBS record. Each patient is assigned a unique identifier, making it possible to uniquely identify patients without breaching confidentiality.

The QLDS includes only 70% of all hospitalisations in Queensland, because patients who did not present a Medicare card have been excluded. As a consequence, certain population subgroups are under-represented in the hospital data. The biases associated with these data are described in a previous report [16]. The number of patients in the QLDS is 1,176,294.

#### 3.2 Population Selection

A patient is flagged as a CRC patient if they have ever had a hospital separation between July 1995 and June 1999 with a diagnosis indicating CRC. The ICD9 (The International Classification of Diseases, 9th Revision) codes included are those beginning with 153 (for malignant neoplasm colon) or 154 (malignant neoplasm of rectum/anus). All ten diagnosis flags in hospital separation data are considered. There are 8,104 such patients. In our analysis, the CRC patients are classified into class 1, all the other patients are classified into class 0.

#### 3.3 Feature Selection

Table 1 lists the features selected for the study. The postcode is based on the patients' MBS records. Those patients who do not have any MBS records or their

postcode does not fall in Queensland have the field value “NO” as a missing value. For the patients who have more than one MBS record, the majority value is used to decide the value of postcode. The Seifa (Social Economic Index for Areas) data are mapped from postcodes according to the 1996 Australian Census data. The Aria (Accessibility/Remoteness Index of Australia) data are derived from postcodes to reflect the accessibility to health care facilities.

**Table 1.** Features selected for the study

Feature	Description	Data Type
Linkid	Encrypted link id	ID variableSymbol
Gender	m: male, f: female	Binary
Age	Age at 1995	Integer
Age Group	Discrete age group	00-43,44-53,54-63,64-73,74-00
Postcode	Postcode of Patient	categorical
Aria Continuous	Access to health facility	Continuous
Aria Discrete	Access to health facility	HA, A, MA,R, VR
Seifa Continuous	Postcode's average household income <sup>1</sup>	Continuous
Seifa Discrete	Postcode's average household income	High, Medium, Low
Consultation Continuous	Number of physician consultations	Continuous
Consultation Discrete	Number of physician consultations	High, Medium, Low
Diagnostic Continuous	Number of diagnostic items in MBS	Continuous
Diagnostic Discrete	Number of diagnostic items in MBS	High, Medium, Low
Procedure continuous	Number of procedure items in MBS	Continuous
neoplasm	neoplasm flag	0,1
diabetes	diabetes flag	0,1
mental	mental flag	0,1
circulatory	circulatory flag	0,1
heart	heart flag	0,1
respiratory	respiratory flag	0,1
asthma	asthma flag	0,1
muscolo	muscolo flag	0,1
Class	0:non-crc 1:crc	Binary

Consultation and diagnosis record the average number of MBS services per year. This is calculated for the period prior to the first CRC hospital event for CRC patients and for the entire five years (1996–1999) for non-CRC patients. Consultation is discretised to Low ( $c < 4.8$ ), Medium ( $4.8 \leq c < 9.2$ ) and High( $c \geq 9.2$ ). Diagnostic is discretised to Low ( $d < 2.0$ ), Medium ( $2.0 \leq d < 5.43$ ) and High( $d \geq 5.43$ ). These cutoff values are chosen based on results from running an association algorithm (Magnum Opus). The discretised Seifa values are Low ( $s \leq 856.86$ ), Medium( $856.86 < s \leq 1032.15$ ) and High( $s > 1032.15$ ) so that the population of Queensland has 25% belonging to High, 50% to Medium and 25% to Low.

## 4 Association Rule Discovery

### 4.1 Method

The aim of association analysis is to discover the association between available variables and the colorectal cancer prevalence. Magnum Opus was first applied to the whole population in the QLDS. Magnum Opus is an ease to use association rule discovery tool with excellent flexibility. It finds rules from both transaction data and attribute-value data efficiently [14]. It can discretise the numeric attributes automatically.

<sup>1</sup> Year 2000 survey result from Australian Bureau of Statistics

## 4.2 Feature Selection

The selected features used for analysis are listed as follows:

- Gender: m, f
- Age: numeric 3
- AriaDis: categorical
- Seifa: numeric 3
- Consultation: numeric 3
- Diagnostic: numeric 3
- Procedure: numeric 3
- Class: 0, 1

All numeric features are discretised into three sub-ranges, each of which contains approximately the same number of cases.

## 4.3 Results for All Patients

A rule has two parts: a Left Hand Side (LHS) and a Right Hand Side (RHS). The strength of a rule is the proportion of examples covered by the LHS of the rule that are also covered by the RHS. The lift of a rule is the strength divided by the RHS coverage proportion. This indicates how much more frequent the RHS is if the LHS occurs than normal. Table 2 shows a part of the association rules sorted by lift in descending order. Our observations are as follows.

**Table 2.** Part of the rules identified by Magnum Opus on all patients

Rule No	LHS	RHS(Class 1)	Lift
1	Gender=m Age > 50 AriaDis=HA Consultation < 5.80	806	4.75
2	Age > 50 AriaDis=HA Consultation < 5.80	1299	4.61
3	Age > 50 Consultation < 5.80 $2.40 \leq \text{Diagnostic} \leq 6.40$	767	4.55

- Rule 1 is interesting, covering about 10% of the 8,104 CRC patients. This group of patients include males aged above 50, with high accessibility to health facilities and having consultation counts less than 5.8. Patients identified by the rule are 4.75 times more likely to have CRC than the general population.
- Rule 2 is a more general rule covering 16% of the CRC population and retaining a lift of 4.61. Rule 3 is similar.
- These rules all suggest that older patients (50+) with accessibility to health care facilities but low utilisation rates are more than four times more likely than the general population to develop colorectal cancer.

#### 4.4 Results for Patients Over 44

Since most CRC patients are older than 40, we selected patients over 44 years of age to form a new dataset for analysis. Magnum Opus was applied to this dataset with selected features and parameters for discretisation as above. Table 3 shows a part of the association rules sorted by lift in descending order. Our observations are as follows.

**Table 3.** Part of the rules identified by Magnum Opus on all patients over 44

Rule No	LHS	RHS(Class 1)	Lift
1	$55 \leq \text{Age} \leq 68$ Consultation < 8.00 circulatory=1 heart=0	565	2.82
2	AriaDis=HA Consultation < 8.00 respiratory=1 asthma=0	486	2.55
3	$55 \leq \text{Age} \leq 68$ Consultation < 8.00 circulatory=1	714	2.41
4	AriaDis=HA Consultation < 8.00 respiratory=1	572	2.35
5	Consultation < 8.00 heart=0 respiratory=1	711	2.28
6	Consultation < 8.00 respiratory=1 asthma=0	726	2.22
7	AriaDis=HA Consultation < 8.00 circulatory=1 heart=0	760	2.20
8	Gender=m $55 \leq \text{Age} \leq 68$ Consultation < 8.00 muscolo=0	906	2.15
9	$55 \leq \text{Age} \leq 68$ AriaDis=HA Consultation < 8.00 muscolo=0	863	2.12
10	Consultation < 8.00 circulatory=1 heart=0 muscolo=0	1052	2.10

- Patients aged between 55 and 68 with circulatory disease and a low utilisation of consultations are more than twice as likely than the general population to have colorectal cancer.
- Patients with circulatory disease and a low utilisation of consultations living in regions highly accessible to health care facilities are more than twice as likely than the general population to have colorectal cancer.

## 5 Scalable Cluster Analysis

### 5.1 Method

Clustering is one of the most widely used techniques in data mining. It is used to reveal patterns in data that can be extremely useful to data analysts. The task of clustering is to partition a data set into clusters in such a way that the data

records within each cluster are more similar among themselves than data records in other clusters [5, 7]. A scalable clustering system, the computational time of which grows linearly or sub-linearly with the number of data records, bridges the gap between the limited computational resources and large databases [8, 6].

We employed a scalable clustering algorithm, BIRCH [17], to identify the groups of patients who are more likely to suffer from CRC. First we normalised each continuous attribute into the interval [0,1]. Then BIRCH with default setting was used to generate 100 clusters based on these continuous attributes. After that, CRC patients within each cluster was used to identify high risk clusters in comparison with the whole data set. For example, the lift is defined as the proportion of CRC patients covered by a cluster divided by the proportion of non-CRC patients covered by this cluster. It roughly indicates to what degree this cluster of people are more likely to suffer from CRC than the whole population. To further understand the relationship between one cluster indicator and the CRC class, we made a bivariate tabular analysis over them and got its Chi-Square value. The Chi-square value also indicates how much more frequently the cluster of people suffer from CRC than the whole population. The clusters that have less than 200 patients are left out since they are too small compared with the whole data set.

**Table 4.** Typical clusters with high risk for CRC patients identified from all the 1,176,294 patients

Cluster ID	Age	Aria-Con	Seifa	Consultation	Diagnostic	Class 1	Coverage		Lift	Chi-Square
							Cardinality	%		
0	81.7	0.12	859.6	11.9	5.7	45	1623	0.144	4.17	106.82
82	71.7	4.23	967.2	13.2	8.5	236	9485	0.844	3.73	469.70
98	71.5	4.97	1014.6	11.8	7.2	38	1696	0.151	3.35	62.19
12	79.4	0.55	965.7	16.0	8.5	821	38625	3.437	3.18	1257.08
43	65.4	0.02	1048.5	43.8	62.1	62	2934	0.261	3.16	90.89
63	77.5	2.84	963.8	13.7	7.9	377	18120	1.613	3.11	543.53
46	78.2	0.13	1164.3	15.1	9.0	65	3146	0.280	3.09	91.21
39	78.0	5.92	950.0	11.9	6.7	68	3298	0.293	3.08	95.05
72	67.7	0.32	969.3	15.1	9.5	1293	62716	5.581	3.08	1907.42
83	64.7	11.46	943.8	8.2	4.2	15	728	0.065	3.08	20.89
37	67.9	2.74	882.5	10.8	6.3	58	2820	0.251	3.07	80.64
55	62.1	7.89	922.1	9.8	6.1	23	1128	0.100	3.04	31.39
86	78.2	0.06	1049.2	16.2	9.3	559	27477	2.445	3.04	777.53
6	78.4	10.64	923.3	9.3	3.5	19	948	0.084	2.99	25.05
95	65.6	0.11	1045.6	14.4	9.1	750	37627	3.349	2.97	1010.54
53	80.8	3.54	897.7	11.9	5.3	45	2345	0.209	2.86	54.33
79	60.5	3.09	995.7	11.4	7.8	266	14073	1.252	2.82	314.40
47	61.1	2.64	941.2	11.3	7.6	424	22693	2.020	2.79	492.82
76	64.7	4.15	907.8	10.4	6.3	63	3396	0.302	2.76	70.88
97	60.9	10.17	1020.7	6.5	3.9	12	655	0.058	2.73	13.11
58	72.9	1.70	1020.4	12.9	7.8	54	3011	0.268	2.67	56.43
75	64.7	5.79	953.2	10.4	7.0	97	5566	0.495	2.59	95.29
78	55.9	0.36	869.4	11.6	7.2	72	4145	0.369	2.59	70.12
13	78.0	8.12	921.3	9.8	4.2	7	411	0.037	2.53	6.50
21	60.8	0.16	1168.3	12.3	8.6	102	6417	0.571	2.36	80.67
74	62.4	10.47	892.7	8.7	5.0	24	1521	0.135	2.35	18.56
66	57.9	1.76	1035.7	10.7	6.9	43	3191	0.284	2.00	21.64

## 5.2 Feature Selection

The selected features are listed as following.

- Age: numeric
- AriaCon: numeric
- Seifa: numeric
- Consultation: numeric
- Diagnostic: numeric
- Class: 0, 1

## 5.3 Clusters for All Patients

We first applied BIRCH to generate 100 clusters for all the 1,176,294 patients. It took about 8.19 seconds in total and about 52,608 patients were not clustered and viewed as outliers.

Table 4 lists typical clusters with high proportions of CRC patients. The clusters are listed in descending order with respect to their lift. The clusters with lift less than 2.0 are omitted from the table. Each row indicates an interesting cluster described by a cluster centroid. For example, as listed in the first row of Table 4, Cluster 0 has a centroid of Age: 81.7, Aria: 0.12, Seifa: 859.6, Consultations: 11.9, and Diagnostics: 5.7. There are 1,623 patients in the cluster, and 45 CRC patients. The lift is 4.17, i.e., the patients within the cluster are 4.17 times more likely to suffer from CRC. The estimated  $\chi^2$  is 106.82. It indicates that this cluster of patients are significantly more likely to suffer from CRC, compared with the whole data set. Similar interesting results can be found in Table 4.

## 5.4 Clusters for Patients Over 44

We also conducted cluster analysis on the patients over 44 years of age. BIRCH took about 2.39 seconds to generate 100 clusters from the 453,645 patients and generated 27,955 outliers.

Table 5 lists some typical clusters with high proportions of CRC patients from these old patients. They are sorted by lift in descending order, while those with lift less than 1.30 are omitted. A typical example is Cluster 31 as listed in Table 5. Its cluster centre is Age: 65.1, Aria: 5.41, Seifa: 896.3, Consultations: 41.3, and Diagnostics: 54.0. There are 284 patients in the cluster, and 11 CRC patients. The lift is 2.50 while the estimated  $\chi^2$  is 12.12. Again, this cluster of patients are significantly different from other patients over 44 years of age. Similar results can be observed from other clusters.

# 6 Association Classification

## 6.1 Method

The association classification algorithm developed in [9] generates the optimal class association rule set. The experimental results in [9] show that the optimal class rule set achieves a very high classification accuracy.

**Table 5.** Typical clusters with high risky of CRC patients on 453,645 patients elder than 44

Cluster ID	Age	Aria-Con	Seifa	Consu-Itation	Diag-nostic	Class 1	Coverage		Lift	Chi-Square
							Cardinality	%		
31	65.1	5.42	896.3	41.3	54.0	11	284	0.063	2.50	12.12
88	62.9	7.17	886.8	26.6	35.0	11	296	0.065	2.39	146.43
29	75.2	8.40	922.0	37.7	49.3	7	216	0.048	2.08	81.51
30	81.1	3.64	861.0	25.8	33.9	13	423	0.093	1.97	387.30
65	69.1	11.03	938.3	33.8	44.1	12	416	0.092	1.84	1227.02
71	68.2	4.98	1016.0	31.1	40.6	34	1295	0.285	1.67	452.60
2	73.3	0.31	1199.2	14.5	19.0	26	996	0.220	1.66	376.44
76	72.9	4.74	907.5	31.2	40.7	34	1338	0.295	1.62	744.56
37	65.3	0.14	887.2	28.0	36.6	152	6040	1.331	1.60	850.28
81	68.0	3.62	981.7	38.2	49.9	126	5133	1.132	1.56	460.45
67	74.5	3.72	981.9	40.5	52.8	85	3487	0.769	1.55	69.91
4	82.0	5.93	952.1	26.3	34.3	43	1774	0.391	1.54	19.95
13	79.0	0.06	866.1	35.7	46.7	49	2061	0.454	1.51	25.83
26	81.9	3.59	991.3	35.5	46.4	82	3540	0.780	1.47	40.44
15	66.1	0.31	1197.0	31.9	41.7	20	867	0.191	1.46	89.91
86	73.6	0.09	1133.6	37.3	48.7	22	985	0.217	1.42	12.61
16	71.5	0.39	966.4	41.0	53.5	355	15943	3.514	1.41	19.28
79	76.1	0.66	945.5	37.7	49.2	284	12816	2.825	1.41	63.12
90	80.8	4.10	933.4	33.6	43.9	39	1766	0.389	1.40	838.58
78	75.9	0.10	1044.1	28.3	36.9	420	19294	4.253	1.38	10.81
39	80.6	0.31	1198.2	30.5	39.8	11	514	0.113	1.36	103.71
44	65.6	2.58	945.7	35.2	45.9	212	9921	2.187	1.35	5.77
46	77.3	2.69	944.4	36.0	46.9	98	4628	1.020	1.34	18.93
94	61.3	3.12	997.6	32.7	42.6	160	7569	1.668	1.34	7.69
34	69.1	0.09	1049.0	34.5	45.0	424	20059	4.422	1.34	84.16
82	71.0	2.82	940.2	32.5	42.4	188	9016	1.987	1.32	4.04
54	58.4	0.34	869.3	20.7	27.1	40	1942	0.428	1.30	23.19
51	73.5	1.75	1032.9	33.5	43.7	34	1654	0.365	1.30	6.46
27	66.3	3.24	887.3	37.4	48.8	24	1168	0.257	1.30	47.37
74	81.7	10.53	903.4	32.3	42.3	6	292	0.064	1.30	22.58

However, our dataset has very unbalanced classes. Our main interest is in finding rules (or cohorts) which lead to higher occurrences of colorectal cancer patients than the average occurrence. As a result, the original algorithm has been modified to increase classification accuracy of class 1 patients. The modification is that, instead of using the minimum global support as a criterion for rules to be included, local support is introduced to find the rules describing the small class (class 1). *Local Support* is defined by Equation 1.

$$lsup(A \rightarrow c) = \frac{sup(A \rightarrow c)}{sup(c)} \quad (1)$$

Here  $sup(c)$  and  $sup(A \rightarrow c)$  represent the support (or proportion or relative frequency) of class  $c$  in the whole population and the support of pattern  $A$  in class  $c$  respectively. The algorithm will identify rules which give high “lift” values for class 1. Lift is defined in Equation 2.

$$lift(A \rightarrow c) = \frac{lsup(A \rightarrow c)}{sup(A)} \quad (2)$$

## 6.2 Results for All Patients

Example rules identified are listed in Table 6. Rule 1 identifies the patients who have the following characteristics:

- Aged between 64 and 73.
- In area highly accessible to medical facilities.
- Small number of doctor's consultations is low
- No heart and musculoskeletal disease.

There are a total of 273 CRC patients in this group. The lift of the group is 6.74. It implies that the individuals who have these characteristics are 6.74 times likely to have CRC than general population. Rule 62 indicates that for male population, living in highly accessible area with circulatory and respiratory disease, but no heart and asthma disease, the likelihood of CRC is 5.05. Rule 62, 79 and 91 all suggest that CRC is correlated with circulatory and respiratory disease.

**Table 6.** Part of the rules identified by association classification algorithm

Rule No	Rule	Class 1	Lift
1	Age = 64-73 Aria = HA Consultation = Low heart = 0 muscolo = 0	273	6.74
3	Age = 54-63 Aria = HA Consultation = Low heart = 0 muscolo = 0	317	6.28
10	Age = 64-73 Consultation = Medium diabetes = 0 circulatory = 1 heart = 0	255	6.19
62	Gender = m Aria = HA mental = 0 circulatory = 1 heart = 0 respiratory = 1 asthma = 0	260	5.05
66	Gender = m Age = 64-73 heart = 0 respiratory = 1 asthma = 0	269	4.97
79	Age = 74-00 circulatory = 1 heart = 0 respiratory = 1 muscolo = 0	278	4.89
91	Consultation = Low circulatory = 1 respiratory = 1 asthma = 0	283	4.77

## 7 Discussion and Conclusion

Three different data mining techniques have been used to explore for possible factors which may contribute to colorectal cancer. The analysis was performed to two populations in this study. The first population comprises the those population who have developed colorectal cancer during the period of study. The second population consists of patients who have not developed colorectal cancer. The analysis explored the main differences between the two populations to identify potential factors which might lead to high risk of colorectal cancer. Each technique was applied to the two datasets with demographic and socio-economical variables and variables extracted from patients' health care history.

Most of the interesting results are agreeable in terms of high lift value, especially for the results by using association rule and association classification techniques. The results from scalable clustering analysis are not as expressive as those from the former two techniques, but it can efficiently draw a big picture about the characteristics of CRC patients in the background of whole population. These heuristic results from data mining explorations may help health care professionals in identifying areas for further study of the causes and preventative factors of colorectal cancer. All data mining methods identified the following factors as potential risk factors for colorectal cancer:

- Older patients.
- People living near health facilities yet seldom utilising those facilities.
- Patients with respiratory and circulatory diseases.

As mentioned before, limitation of the data (in particular the lack of lifestyle factors including diet, physical exercise, smoking, and drinking) severely limit the detailed analyses. The study is not intended to identify the most important factors leading to colorectal cancer. Rather it can only explore through the variables included in the data sets.

## Acknowledgements

The authors would like to thank their colleagues, including Ross Sparks and Jisheng Cui, as well as Jiuyong Li of University of South Queensland and the anonymous reviewers for their comments and suggestions. The authors acknowledge the Commonwealth Department of Health and Ageing, and the Queensland Department of Health for providing data for this research.

## References

1. Colorectal cancer: The importance of prevention and early detection. Division of Cancer Prevention and Control, National Center for Chronic Disease Prevention and Health Promotion, Centers for Disease Control and Prevention, U.S. Department of Health and Human Services, 2004.

2. J. Chen, H. He, G. Williams, and H. Jin. Temporal sequence associations for rare events. In *Proceedings of 8th Pacific-Asia International Conference on Knowledge Discovery and Data Mining (PAKDD), Lecture Notes in Computer Science (LNAI 3056)*, pages 235–239, Sydney, Australia, May 2004.
3. K. J. Cios and G. W. Moore. Uniqueness of medical data mining. *Artificial Intelligence in Medicine*, 26(1-2):1–24, 2002.
4. L. Gu, J. Li, H. He, G. Williams, S. Hawkins, and C. Kelman. Association rule discovery with unbalanced class. In *Proceedings of the 16th Australian Joint Conference on Artificial Intelligence (AI03), Lecture Notes in Artificial Intelligence*, Perth, Western Australia, December 2003.
5. J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, San Francisco, CA, USA, 2001.
6. H.-D. Jin, K.-S. Leung, M.-L. Wong, and Z.-B. Xu. Scalable model-based clustering of large data sets: Working on clustering features. *Pattern Recognition*, Oct. 2004. In press.
7. H.-D. Jin, W. Shum, K.-S. Leung, and M.-L. Wong. Expanding self-organizing map for data visualization and cluster analysis. *Information Sciences*, 163:157–173, Jun. 2004.
8. H.-D. Jin, M.-L. Wong, and K.-S. Leung. Scalable model-based clustering by working on data summaries. In *Proceedings of Third IEEE International Conference on Data Mining (ICDM 2003)*, pages 91–98, Melbourne, Florida, USA, Nov. 2003.
9. J. Li, H. Shen, and R. Topor. Mining the optimal class association rule set. *Knowledge-based Systems*, 15(7):399–405, 2002.
10. D. McClisha, L. Penberthyb, and A. Pughc. Using medicare claims to identify second primary cancers and recurrences in order to supplement a cancer registry. *Journal of Clinical Epidemiology*, 56:760–767, 2003.
11. R. B. Rao, S. Sandilya, R. S. Niculescu, C. Germond, and H. Rao. Clinical and financial outcomes analysis with existing hospital patient records. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 416 – 425, 2003.
12. J. Roddick, P. Fule, and W. Graco. Exploratory medical knowledge discovery : Experiences and issues. *SIGKDD Exploration*, 5(1):94–99, 2003.
13. A. E. Smith and S. S. Anand. Patient survival estimation with multiple attributes: adaptation of coxs regression to give an individuals point prediction. In *Proceedings of European Conference in Artificial Intelligence in Intelligent Datamining in Medicine & Pharmacology*, pages 51–54, Berlin, 2000.
14. G. I. Webb. Efficient search for association rules. In *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 99–107, 2000.
15. G. Williams, D. Vickers, R. Baxter, S. Hawkins, C. Kelman, R. Solon, H. He, and L. Gu. The Queensland Linked Data Set. Technical Report CMIS 02/21, CSIRO Mathematical and Information Sciences, Canberra, 2002.
16. G. Williams, D. Vickers, C. Rainsford, L. Gu, H. He, R. Baxter, and S. Hawkins. Bias in the Queensland Linked Data Set. Technical Report CMIS 02/117, CSIRO Mathematical and Information Sciences, Canberra, 2002.
17. T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: A new data clustering algorithm and its applications. *Data Mining and Knowledge Discovery*, 1(2):141–182, 1997.

# Efficiently Identifying Exploratory Rules’ Significance

Shiying Huang and Geoffrey I. Webb

School of Computer Science and Software Engineering  
Monash University  
Melbourne VIC 3800, Australia  
{Shiying.Huang, Geoff.Webb}@infotech.monash.edu.au

**Abstract.** How to efficiently discard potentially uninteresting rules in exploratory rule discovery is one of the important research foci in data mining. Many researchers have presented algorithms to automatically remove potentially uninteresting rules utilizing background knowledge and user-specified constraints. Identifying the significance of exploratory rules using a significance test is desirable for removing rules that may appear interesting by chance, hence providing the users with a more compact set of resulting rules. However, applying statistical tests to identify significant rules requires considerable computation and data access in order to obtain the necessary statistics. The situation gets worse as the size of the database increases. In this paper, we propose two approaches for improving the efficiency of significant exploratory rule discovery. We also evaluate the experimental effect in impact rule discovery which is suitable for discovering exploratory rules in very large, dense databases.

## Keyword

Exploratory rule discovery, impact rule, rule significance, interestingness measure

## 1 Introduction

Exploratory rule discovery techniques seek multiple models which are able to efficiently describe the potentially interesting inter-relationships among attributes in a database. Searching for multiple models instead of a single model often results in numerous spurious or uninteresting rules.

How to automatically discard statistically insignificant rules has been an important issue in research of exploratory rule discovery. Several papers have been devoted to this topic. Bay and Pazzani [4], Liu et. al [10] and Webb [15], developed techniques for identifying insignificant rules with qualitative attributes only (or discretized quantitative attributes). Aumann and Lindell [2] and Huang and Webb [8] both did research on exploratory rule significance with undiscrctized quantitative attributes as consequent.

When filtering insignificant exploratory rules regarding quantitative attributes, the rule discovery systems have to go through the database several times so as

to collect the necessary parameters for the significance test. Moreover, considerable CPU time has to be spent on data access and looking for the set of records which is covered by the antecedent of a rule. For example, it has been shown by Huang and Webb [8] that the time spent for discovering the top 1000 significant impact rules is on the whole much more than that spent on discovering the top 1000 impact rules without using any filter, especially when most of the top 1000 impact rules are insignificant. A technique for improving the efficiency of the insignificance filter is presented in the same paper by introducing the triviality filter. The anti-monotonicity of triviality was utilized to effectively prune the search space.

There is an immediate need for improving the efficiency of the insignificance filter for distributional-consequent exploratory rule discovery, even after the introduction of the triviality filter. In this paper, we propose two approaches for efficiency improving in exploratory rule discovery, which can result in substantial reduction of the computation for discovering significant rules. Although the demonstration is done on impact rule discovery, these techniques can also be recast for other exploratory rule discovery tasks.

The paper is organized as follows: In section 2, we introduce the concept and notations of exploratory rule discovery. Existing techniques for discarding insignificant exploratory rules is introduced in section 3, followed by the brief description of impact rule discovery in section 4. The techniques for improving the efficiency are presented in section 5. In section 6, we provide experimental results and evaluations. Conclusions are drawn in section 7.

## 2 Exploratory Rule Discovery

Traditional machine learning systems discover a single model from the available data that is expected to maximize the accuracy or some other specific measures of performance on unknown future data. Predictions or classifications are then done on the basis of this single model [15]. Examples include the decision tree [12], the decision rules [11], and the Naive-Bayes classifier. However, alternative models exist that perform equally well as those which are selected by the systems. Thus, it is not always sensible to choose only one of the “best” models in some cases. The criteria for deciding whether a model is best or not also varies with the context of application. Exploratory rule discovery techniques are proposed to overcome this problem by searching for multiple models which satisfy certain constraints and presenting all these models to the user. Thus, the users are provided with alternative choices. Better flexibility is achieved herewith.

Exploratory rule discovery techniques [8] are classified into propositional rule discovery which seeks rules with qualitative attributes or discretized quantitative attributes only and distributional-consequent rule discovery which seeks rules with quantitative attributes as consequent. The status or performance such quantitative attributes are described with their distributions. *Association rule discovery* [1], *contrast sets discovery* [4] are examples of propositional exploratory rule discovery, while *impact rule discovery* [13] and *quantitative association rule*

*discovery* [2] both belong to the class of distributional-consequent rule discovery. It is argued that distributional-consequent rules are able to provide better descriptions of the interrelationship between quantitative attributes and qualitative attributes.

Here are some notions of exploratory rule discovery that we are to use in this paper:

1. A *dataset* is a finite set of *records*
2. For propositional rule discovery, a *record* is an element to which we apply Boolean predicates called conditions, while for distributional-consequent rule discovery, a record is a *pair*  $\langle c, v \rangle$ , where  $c$  is the nonempty set of Boolean conditions, and  $v$  is a set of values for the quantitative variables in whose distribution the users are interested.
3. A rule is in the form of  $A \rightarrow C$ . For propositional rules, both  $A$  and  $C$  are conjunctions of Boolean conditions. The status of such rule is described by interestingness measures like the *support* and the *confidence*. Contrarily, for distributional-consequent rule discovery,  $A$  is a conjunction of Boolean conditions while  $C$  is a nonempty set of target quantitative variables in which the users are interested. The quantitative variables are described by distributional statistics. We prefer using  $A \rightarrow target$  to denote a distributional-consequent rule instead, for the purpose of avoiding confusion.
4. Rule  $A \rightarrow C$  is a parent of  $B \rightarrow C$  if  $A \subset B$ . If  $|A| = |B| - 1$  than the second rule is a direct parent of the first one, otherwise, it is a grandparent of the first rule.
5. We use the notion  $coverset(A)$ , where  $A$  is a conjunction of conditions, to represent the set of records that satisfy the condition (or set of conditions)  $A$ . If a record  $x$  is in  $coverset(A)$ , we say that  $x$  is *covered* by  $A$ . If  $A$  is  $\emptyset$ ,  $coverset(A)$  includes all the records in the database.
6.  $Coverage(A)$  is the number of records covered by  $A$ .  $coverage(A) = |coverset(A)|$ .

### 3 Insignificant Exploratory Rules

As is mentioned before, exploratory rule discovery searches for multiple models in a database, and may lead to discovering spurious or uninteresting rules. How to decrease the number of resulting rules becomes a problem of concern. One approach is up to the users to define a suitable set of constraints which may be utilized so that the algorithm can automatically discard some potentially uninteresting rules. Another approach is to perform comparison within resulting rules, so as to present the users with a more compact set of models. Techniques regarding automatically removing potentially uninteresting rules are summarized by Huang and Webb [8].

#### 3.1 Improvement

Filtering insignificant rules using statistical tests is one of the interesting topics of research. By using this technique we perform significance tests among rules

and discard those happen to appear interesting only by chance. To provide a clear idea of insignificant rules, we will at first introduce the concept of rule *improvement* defined by Bayardo et al. [5]. *Confidence improvement* which is used as an example, defined a minimum improvement in confidence that a propositional rule must exhibit in order to be regarded as potentially interesting:

$$\text{imp}(A \rightarrow C) = \min(\forall A' \subset A, \text{confidence}(A \rightarrow C) - \text{confidence}(A' \rightarrow C))$$

It is argued that setting a minimum improvement is desirable in discarding potentially uninteresting exploratory rules. However, the values used for comparison are derived from samples instead of from the total population. There is the problem that the observed improvement provides only an estimate of the true improvement, and if no account is taken of the quality of that estimate, so it is likely to result in poor decisions.

Rule filtering techniques regarding the significance of rules concern about the statistical significance of the improvement, rather than the values of interestingness measures. Statistical tests are done with resulting rules and those within expectation (or without enough surprisingness) are automatically removed. Such techniques may lead to type-1 error, which result in accepting spurious or uninteresting rules and type-2 error, which result in rejecting rules that are not spurious. A technique for statistically sound exploratory rule discovery is proposed by Webb [15] using a holdout set to validate the resulting rules.

### 3.2 Statistical significance of rules

Chi-square test is a widely used test for identifying propositional rule independence. Liu et al. [10] did research on association rules with a fixed attribute as consequent. They used a chi-square test to decide whether the antecedent of a rule is independent from its consequent or not, accepting only rules whose antecedent and consequent are positively correlated, thus, discarding rules which happen to appear interesting by chance. The rules discarded by using an independent test are referred to as insignificant rules.

Consider the following Boolean-consequent rules:

$$A \rightarrow C[\text{support} = 60\%, \text{confidence} = 90\%]$$

$$A \& B \rightarrow C[\text{support} = 45\%, \text{confidence} = 91\%]$$

$$A \& D \rightarrow C[\text{support} = 46\%, \text{confidence} = 70\%]$$

There is a high possibility that the conditions  $B$  and  $C$  are conditionally independent given  $A$ , thus the second rule provides little interesting information. According to Liu et al., the third rule does not bear interesting information, either. It should also be discarded, because the condition  $D$  is negatively correlated to condition  $C$ , given  $A$ . Bay and Pazzani [4] also made use of Chi-square test to decide the significance of *contrast sets*. Webb [15] proposed a statistically

sound technique for filtering insignificant rules, using the Fisher exact test and a hold out set.

Aumann and Lindell [2] and Huang and Webb [8] both proposed ideas for filtering insignificant distributional-consequent exploratory rules. In this paper, we use the definition proposed by the latter.

**Definition 1. *significant impact rule*** *An impact rule  $A \rightarrow target$  is significant if the distribution of its target is significantly improved in comparison with the target distribution of any of its direct parents'. The measure for the target distribution can be the mean, the variance etc.*

$$significant(A \rightarrow target) = \forall x \in A, dist(coverset(A))$$

$$\gg dist(coverset(A - x) - coverset(A))^1$$

*An impact rule is insignificant if it is not significant.*

Definitions of insignificant propositional exploratory rules are provided by Liu et al. [10] and Bay and Pazzani [4].

In this paper, the mean of the target attribute over  $coverset(A)$  is used as the interestingness measure to be compared for the impact rule. Statistical test is done to decide whether the target means of two samples are significantly different from each other.

#### 4 K-Most-Interesting Impact Rule Discovery and Notations

The impact rule discovery algorithm we adopt is based on the OPUS [14] algorithm, which enable the successfully discovery of the top  $k$  impact rules that satisfy a certain set of constraints.

We characterized the terminology of k-most-interesting impact rule discovery to be used in this paper as follows:

1. An impact rule is in form of  $A \rightarrow target$ , while the target is describe by the following measures: *coverage*, *mean*, *variance*, *maximum*, *minimum*, *sum* and *impact*.
2. *Impact* is a interestingness measure suggested by Webb [13]<sup>2</sup>:  $impact(A \rightarrow target) = (mean(A \rightarrow target) - \overline{targ}) \times coverage(A)$ .
3. An k-most-interesting impact rule discovery task is a 7-tuple:  
 $KMIIRD(\mathcal{C}, \mathcal{T}, \mathcal{D}, \mathcal{M}, \lambda, \mathcal{I}, k)$ .

$\mathcal{C}$ : is a nonempty set of Boolean conditions, which are the set of available conditions for impact rule antecedents.

<sup>1</sup> The token “ $\gg$ ” is used to denote **significantly improved**, and  $dist(\mathcal{R})$  is used to represent the distribution of the target variable over the set of records  $\mathcal{R}$ .

<sup>2</sup> In this formula,  $mean(A \rightarrow target)$  denotes the mean of the *targets* covered by  $A$ , and  $coverage(A)$  is the number of the records covered by  $A$ .

Algorithm: OPUS\_IR\_Filter(Current, Available,  $\mathcal{M}$ )

```

1. SoFar := {}
2. FOR EACH P in Available
  2.1 New := Current  $\cup$  P
  2.2 IF New satisfies all the prunable constraints in  $\mathcal{M}$  except the nontrivial [8]
      constraint THEN
    2.2.1 IF any direct subset of New has the same coverage as New THEN
      New  $\rightarrow$  relevant stats is a trivial rule
      Any superset of New is trivial, so do not access any children of this node,
      go to step 2.
    2.2.2 ELSE IF the mean of New  $\rightarrow$  relevant stats is significantly higher than all its
      direct parents THEN
      IF the rule satisfies all the other non-prunable constraints in  $\mathcal{M}$ 
      THEN record Rule to the ordered rule_list
      OPUS_IR_Filter(New, SoFar,  $\mathcal{M}$ )
      SoFar := SoFar  $\cup$  P
    2.2.3 END IF
  2.3 END IF
3. END FOR
    
```

Table 1. OPUS\_IR\_Filter

$\mathcal{T}$ : is a nonempty set of the variables in whose distribution we are interested.

$\mathcal{D}$ : is a nonempty set of records, which is called the database. A record is a pair  $\langle c, v \rangle$ ,  $c \subseteq C$  and  $v$  is a set of values for  $\mathcal{T}$ .

$\mathcal{M}$ : is a set of constraints. There are two types of constraints *prunable* and *unprunable constraints*. *Prunable constraints* are constraints that you can derive useful bounds for search space pruning and still ensures the completeness of information. Examples include the anti-monotone, the succinct constraints [7], or the convertible constraints [9]. Constraints which are not prunable are *unprunable constraints*

$\lambda$ :  $\{X \rightarrow Y\} \times \{\mathcal{D}\} \rightarrow \mathcal{R}$  is a function from rules and databases to values and defines a interestingness metric such that the greater the value of  $\lambda(X \rightarrow Y, \mathcal{D})$  the greater the interestingness of this rule given the database.

$\mathcal{I}$ : is the set of impact rules that can be derived from  $\mathcal{D}$ , whose antecedents are conjunctions of one or more conditions in  $C$ , whose targets are members of  $\mathcal{T}$ , and which satisfy the constraints in  $\mathcal{M}$ .

$k$ : is a user specified integer number denoting the number of rules in the ultimate solution for this task.

The original algorithm for impact rule discovery with filters are described in table 1. In this table, *current* is the set of conditions, whose supersets are currently being explored. *Available* is the set of conditions that may be added to *current*. By adding every condition in *available* to *current* one by one, we form the antecedent of the *current rule*: *New*  $\rightarrow$  *target*, which will be referred to later as *current.rules*. *Rule\_list* is an ordered list of the top-k interesting rules we have encountered.

## 5 Efficient Identification of Exploratory Rule Significance

### 5.1 Deriving Difference Set Statistics without Data Access

According to the algorithm in table 1 and definition 1, we have to compare the mean of current rule with the means of all its direct parents' in order to decide whether a rule is *significant* or not. The set difference operations necessary for this purpose requires excessive data access and computation. However with the status of current rule and all its parent rules known, we will be able to derive the statistics of the difference sets for performing the significance test, without additional access to the database. The following lemma validates this statement.

**Lemma 1.** *Suppose we are searching for impact rules from a database  $\mathcal{D}$ . If  $A \subset B$ , and  $\text{coverset}(A) - \text{coverset}(B) = \mathcal{R}$ , where  $A$  and  $B$  are both conjunction of conditions,  $\mathcal{R}$  is a set of records from  $\mathcal{D}$ . If the mean and variance of the target attribute over  $\text{coverset}(A)$  and  $\text{coverset}(B)$  are known, as well as the cardinality of both record sets, the mean and variance of the target attribute over set  $\mathcal{R}$  can be derived without additional data access.*

*Proof.* Since  $\text{coverset}(A) - \text{coverset}(B) = \mathcal{R}$ , it is obvious that

$$|\mathcal{R}| = \text{coverage}(A) - \text{coverage}(B) \quad (1)$$

$$\text{mean}(\mathcal{R}) = \frac{\text{coverage}(A) \times \text{mean}(A \rightarrow \text{target}) - \text{coverage}(B) \times \text{mean}(B \rightarrow \text{target})}{|\mathcal{R}|} \quad (2)$$

$$\text{variance}(A \rightarrow \text{target}) = \frac{\sum_{x \in \text{coverset}(A)} (\text{target}(x) - \text{mean}(A \rightarrow \text{target}))^2}{\text{coverage}(A) - 1} \quad (3)$$

$$\text{variance}(B \rightarrow \text{target}) = \frac{\sum_{x \in \text{coverset}(B)} (\text{target}(x) - \text{mean}(B \rightarrow \text{target}))^2}{\text{coverage}(B) - 1} \quad (4)$$

$$\sum_{x \in \text{coverset}(A)} \text{target}(x) = \text{mean}(A \rightarrow \text{target}) \times \text{coverage}(A) \quad (5)$$

$$\sum_{x \in \text{coverset}(B)} \text{target}(x) = \text{mean}(B \rightarrow \text{target}) \times \text{coverage}(B) \quad (6)$$

From 3, 4, 5 and 6 it is feasible to derive the following equation:

$$\begin{aligned} \sum_{x \in \mathcal{R}} \text{target}(x)^2 &= \sum_{x \in \text{coverset}(A)} \text{target}(x)^2 - \sum_{x \in \text{coverset}(B)} \text{target}(x)^2 \\ &= \text{variance}(A \rightarrow \text{target}) \times (\text{coverage}(A) - 1) \\ &\quad + \text{mean}(A \rightarrow \text{target})^2 \times \text{coverage}(A) \\ &\quad - \text{variance}(B \rightarrow \text{target}) \times (\text{coverage}(B) - 1) \\ &\quad - \text{mean}(B \rightarrow \text{target})^2 \times \text{coverage}(B) \end{aligned} \quad (7)$$

$$\sum_{x \in \mathcal{R}} target(x) = \sum_{x \in coverset(A)} target(x) - \sum_{x \in coverset(B)} target(x) \quad (8)$$

Thus,

$$\begin{aligned} variance(\mathcal{R}) &= \frac{\sum_{x \in \mathcal{R}} (target(x) - mean(\mathcal{R}))^2}{|\mathcal{R}| - 1} \\ &= \frac{\sum_{x \in \mathcal{R}} target(x)^2}{|\mathcal{R}| - 1} - \frac{2mean(\mathcal{R}) \sum_{x \in \mathcal{R}} target(x)}{|\mathcal{R}| - 1} + \frac{|\mathcal{R}| mean(\mathcal{R})^2}{|\mathcal{R}| - 1} \end{aligned}$$

Since all the parameters in the right hand side of the equation are already known, we are able to derive all the necessary statistics for doing significance test without accessing the records in  $\mathcal{R}$ . The lemma is proved.

Note: in this proof,  $mean(A \rightarrow target)$  denotes the target mean of the records covered by rule  $A \rightarrow target$ ,  $variance(A \rightarrow target)$  denotes the target variance of the records covered by rule  $A \rightarrow target$ , while  $mean(\mathcal{R})$  denotes the target mean of the records in record set  $\mathcal{R}$ , and  $variance(\mathcal{R})$  represents the target variance of the records in  $\mathcal{R}$ .

By deriving the difference set statistics from the statistics of the *parent rule* and *New*  $\rightarrow target$  in table 1, we are able to save data access and computation for collecting the statistics for performing the significance test, thus improve the efficiency of the search algorithm.

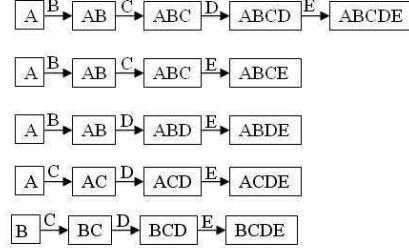
## 5.2 The Circular intersection approach

**Parallel Intersection Approach** According to the definition of significant impact rules, we compare the current rule with all its *direct parents* to identify its significance. In the original OPUS\_IR.Filter algorithm, the procedure described in figure 1 is employed to find the *coverset* of every direct parent of the current rule which is being explored. Each arrow in figure 1 represents an intersection operation. When deciding whether a rule with 5 conditions, namely  $A, B, C, D$  and  $E$  on the antecedent is significant or not, the algorithm have to go through 16 intersection operations! We refer to this approach as the *parallel intersection* approach.

By examining figure 1, we notice that there are considerable overlaps in the *parallel intersection approach*. For example, by using the parallel intersection approach, we have to do the same intersection of  $coverset(A)$  and  $coverset(B)$  three times, when searching for  $coverset(ABCD)$ ,  $coverset(ABCE)$  and  $coverset(ABDE)$ . There must be a way in which two of these operations can be omitted.

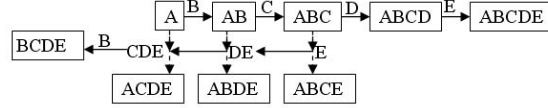
**Circular Intersection Approach** we propose the approach of *circular intersection* which is shown in figure 2<sup>3</sup>. In this approach, intersections are done in

<sup>3</sup> Each dashed arrow in figure 2 and figure 3 points to the outcome of that specific intersection operation and does not represent an actual operation.



**Fig. 1.** The parallel intersection Approach for  $ABCDE$

two stages. Firstly, in the *forward stage*, intersections are done from condition  $A$  to condition  $E$  one at a time, and the results are kept in memory. Then we do intersections from the last condition  $E$  back to the second one  $B$ , which is referred to as the *backward stage*. During the backward stage, the *coverset* of each direct parent of the current rule is found. By introducing the circular intersection approach, the number of intersection operations required for identifying the significance of current rule is reduced to only 10.



**Fig. 2.** The circular intersection approach flow for  $ABCDE$

**Complexity** Using the parallel intersection approach, the number of intersection operations for iterating through all the subsets is:

$$(n - 2) \times n + 1,$$

where  $n$  is the maximum number of conditions on the rule antecedent. The complexity is  $O(n^2)$ .

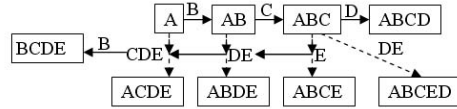
After introducing the circular intersection approach, the intersection operations for iterating through all the subsets are:

$$3n - 5.$$

The complexity is  $O(n)$ . However, practically the difference in running time will not be so dramatic, since we have introduced the triviality filter, which enables

the pruning of the search space. Both the parallel intersection procedure and the circular intersection procedure will probably stop at anytime when it is identified that the current rule is a trivial rule.

The two approaches (the difference set statistics derivation approach and the circular intersection approach) mentioned above can combine with each other so as to achieve higher efficiency. We can save one more intersection operation by introducing the difference set statistics derivation technique in section 5.1. Suppose that we are deciding whether the rule  $A \& B \& C \& D \& E \rightarrow target$  is significant or not. Now that the statistics of one of its parent  $A \& B \& C \& D \rightarrow target$  is known, thus we don't have to derive the statistics of  $cover_{set}(ABCD)$  once again. Hereby, one intersection operation can also be saved by following the procedure shown in figure 3 according to lemma 5.1. The number of necessary



**Fig. 3.** The circular intersection approach for  $ABCDE$  when *current* is  $ABCD$

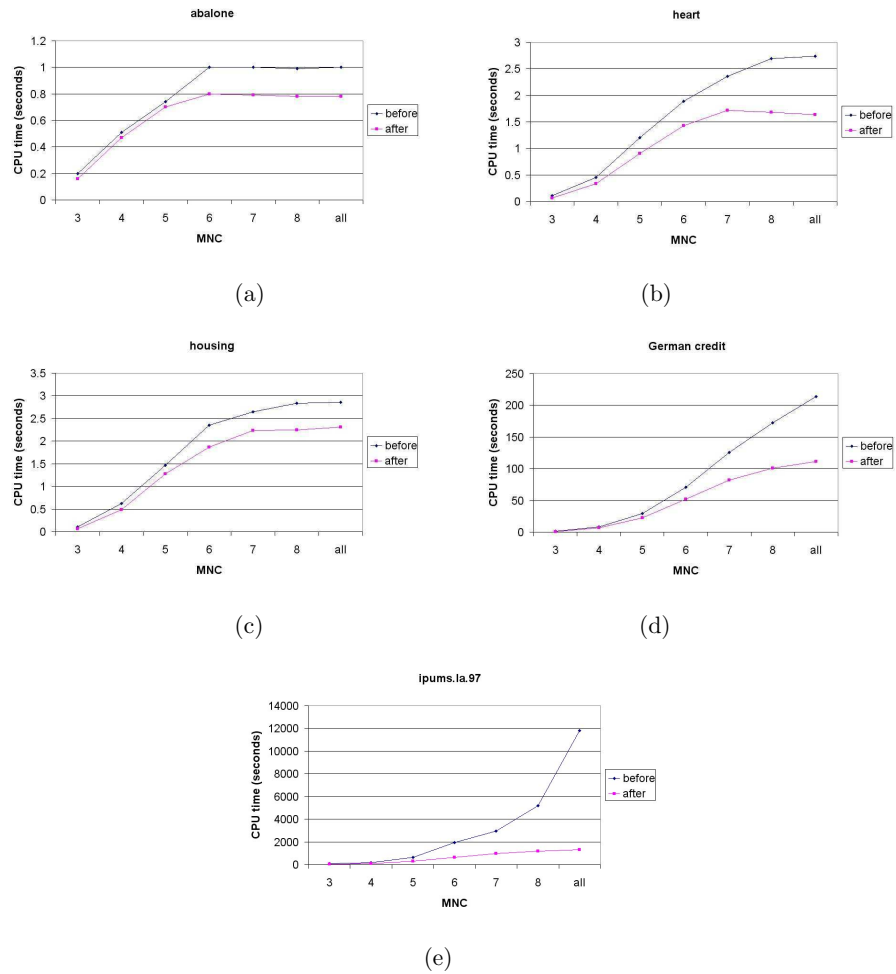
intersection operations is reduced to

$$3n - 6.$$

The new algorithm for impact rule discovery with filters is shown in table 2. In this table, the *parent\_rule* is the corresponding rule for the node whose children we are currently exploring. The antecedent of *parent\_rule* is *current*.

## 6 Experimental Evaluations

In order to explain how the techniques introduced in this paper can practically improve the efficiency of rule discovery, we do our experiments by applying the new algorithm to 10 databases chosen from the UCI Machine Learning repository [6] and the UCI KDD archives [3]. The databases are described in table 3. We applied 3-bin equal-frequency decrepitation to map all the quantitative attributes, except the target attribute, into qualitative ones. The significance level we chose to decide the significance of impact rules is 0.05. The minimum coverage for discovered impact rules is set to 0.01, which is very low. The running time shown in the figures and tables are CPU time spent for the algorithms to search for top 1000 significant impact rules with the highest impact on a computer with two PIII 933MHz processors, 1.5G memory, and 4G virtual memory.



**Fig. 4.** Comparison of Running Time before and after applying data access saving techniques for (a) *abalone*, (b) *heart*, (c) *housing*, (d) *German credit*, and (e) *ipums.la.97*

Algorithm: OPUS\_IR\_Filter(Current, Available, parent\_rule,  $\mathcal{M}$ )

```

1 SoFar :=  $\emptyset$ ;
2 FOR EACH P in Available
2.1 New := Current  $\cup$  P
2.2 IF New satisfies all the prunable constraints in  $\mathcal{M}$  except the nontrivial
    constraint THEN
2.2.1 Derive the statistics of  $cover\_set(Current) - cover\_set(New)$ , according to lemma
    5.1.
2.2.2 IF the mean of  $New \rightarrow target$  is not significantly improved comparing to
     $cover\_set(Current) - cover\_set(New)$  THEN
        go to step 2.2.4;
2.2.3 ELSE use the circular intersection to comparing the mean of  $New \rightarrow target$  with
    the mean of its direct parents other than parent_rule
2.2.3.1 IF the mean  $New \rightarrow target$  is significantly improved comparing to all its
    direct parents THEN
        record  $New \rightarrow target$  to rule_list;
        OPUS_IR_Filter(New, SoFar,  $New \rightarrow target$ );
        SoFar := SoFar  $\cup$  P ;
2.2.3.2 END IF;
2.2.4 END IF;
2.3 END IF;
3 END FOR

```

**Table 2.** Improved OPUS\_IR\_Filter

database	records	attributes	conditions	Target
Abalone	4117	9	24	Shuckedweight
Heart	270	13	40	Max heart rate
Housing	506	14	49	MEDV
German credit	1000	20	77	Credit amount
Ipums.la.97	70187	61	1693	Total income
Ipums.la.98	74954	61	1610	Total income
Ipums.la.99	88443	61	1889	Total income
Ticdata2000	5822	86	771	Ave. income
Census income	199523	42	522	Wage per hour
Covtype	581012	55	131	Elevation

**Table 3.** Basic information of the databases

We ran the program without using the algorithm proposed in table 1 first. For databases *abalone*, *heart*, *housing*, *German credit* and *ipmus.la.97*, which is relatively smaller, we set the maximum number of conditions on the rule antecedent (MNC for short) from 3 to 8, and then run the program with no limit on the MNC. After that, the new algorithm in table 2 is ran according to the same procedure. The CPU time spent for these programs to search for the top 1000 significant impact rules is presented using line charts in figure 4. For *ipmus.la.98*, *ipmus.la.99*, *ticdata2000*, *census income* and *covtype*, which are relatively larger databases, we only ran the programs with MNC set to 3, 4, and 5. The experimental results are listed in table 4.

With *MNC* set to 3, the number of intersection operations required for doing insignificant tests are the same, regardless of whether the circular intersection technique is introduced or not. Thus, the difference in efficiency between the

Database	status	MNC=3	MNC=4	MNC=5
Ipums.la.98	before	74.41	300.47	1860.31
	after	46.15	130.62	482.52
Ipums.la.99	before	750.6	2785.46	9805.81
	after	103.29	312.66	820.72
Ticdata2000	before	116.55	1669.76	10808.03
	after	73.17	1027.33	7946.36
Census-income	before	577.32	2362.53	3781.6
	after	351.56	1054.58	2075.2
Covtype	before	3529.95	11300.45	20686.95
	after	2315.47	9803.97	16987.18

**Table 4.** Time spent (in seconds) for searching for significant rules in databases: *ipums.la.98*, *ipums.la.99*, *ticdata2000*, *census income*, *covtype* before and after the techniques are introduced

algorithms in table 1 and table 2 is caused by applying the data access saving approach which is proposed in section 5.1. For instance, it took the algorithm in table 1 more than 70 seconds to find the top 1000 significant rules in *ipums.la.98* with MNC set to 3, while the time for the algorithm in table 2 to finish the same task is only 57 seconds.

When the MNC is set to a number greater than 3, the trend of increase in running time is much steeper before applying the techniques proposed in section 5 than after. The difference in efficiency increases with the MNC. When there is no limit on the maximum number of conditions on the rule antecedent, the time spent for the new algorithm to search for top 1000 significant impact rules in *ipums.la.97* is less than one sixth of that necessary for the old one. However, the running time is also influenced by other factors including the size of the databases, the number of trivial rules in the top 1000 impact rule, and the number of significant rules.

## 7 Conclusion

The large number of resulting rules has long been a handicap for exploratory rule discovery. Many techniques have been proposed to reduce the set of resulting rules to a manageable size. Removing statistically insignificant rules is one of those techniques that are popular. Such techniques lead to considerable decrease in the resulting number of exploratory rules. However, performing statistical tests to identify the significance of a rule requires considerable data access and computation. We proposed two techniques in this paper, which can improve the efficiency of rule discovery by deriving difference set statistics without additional reference to the data, and by reducing the redundancy of intersection operations. We implemented the techniques in k-most-interesting impact rule discovery, which is suitable for distributional-consequent exploratory rule dis-

covery in very large, dense databases. Experimental results show a substantial improvement in efficiency after applying these techniques.

## References

1. Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami. Mining association rules between sets of items in large databases. In Peter Buneman and Sushil Jajodia, editors, *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216, Washington, D.C., 26–28 1993.
2. Y. Aumann and Y. Lindell. A statistical theory for quantitative association rules. In *Knowledge Discovery and Data Mining*, pages 261–270, 1999.
3. S. D. Bay. The uci kdd archive [<http://kdd.ics.uci.edu>], 1999.
4. S.D. Bay and M.J. Pazzani. Detecting group differences: Mining contrast sets. In *Data Mining and Knowledge Discovery*, pages 213–246, 2001.
5. Roberto J. Bayardo, Jr., Rakesh Agrawal, and Dimitrios Gunopulos. Constraint-based rule mining in large, dense databases. *Data Min. Knowl. Discov.*, 4(2-3):217–240, 2000.
6. C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.
7. J. Han and M. Kamber. *Data mining : concepts and techniques*. Morgan Kaufmann, 2001.
8. Shiyong Huang and Geoffrey I. Webb. Discarding insignificant rules during impact rule discovery in large database, 2004.
9. Jiawei han Jian Pei and Laks V.S. Lakshmanan. Mining frequent itemsets with convertible constraints. In *Proceedings of the 17th International Conference on Data Engineering*, page 433. IEEE Computer Society, 2001.
10. B. Liu, W. Hsu, and Y. Ma. Pruning and summarizing the discovered associations. In *Knowledge Discovery and Data Mining*, pages 125–134, 1999.
11. R. S. Michalski. A theory and methodology of inductive learning. In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach*, pages 83–134. Springer, Berlin, Heidelberg, 1984.
12. J. R. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., 1993.
13. G. I. Webb. Discovering associations with numeric variables. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 383–388. ACM Press, 2001.
14. Geoffrey I. Webb. OPUS: An efficient admissible algorithm for unordered search. *Journal of Artificial Intelligence Research*, 3:431–465, 1995.
15. G.I. Webb. Statistically sound exploratory rule discovery, 2004.

# An Application of Time-Changing Feature Selection

Yihao Zhang<sup>1</sup>, Mehmet A. Orgun<sup>1</sup>, Weiqiang Lin<sup>2</sup>, and Warwick Graco<sup>2</sup>

<sup>1</sup> Department of Computing, I.C.S., Macquarie University Sydney, NSW 2109,  
Australia, Email: {yihao,mehmet}@ics.mq.edu.au

<sup>2</sup> Australian Taxation Office, Canberra ACT 2601,  
Australia, Email: wei.lin,graco.warwick}@ato.gov.au

**Abstract.** This paper describes a time-changing feature selection<sup>1</sup> framework based on hierarchical distribution method for extracting knowledge from time-changing health records. In the framework, we propose three steps for time-changing feature selection. The first step is a qualitative-based search, to find qualitative time-changing features (or, structural time-changing features). The second step performs a quantitative-based search, to find quantitative time-changing features (or, value time-changing features). In the third step, the results from the first two steps are combined to form hybrid search models to select a subset of global time-changing features according to a certain criterion of medical experts. The present application of the time-changing feature selection method involves time-changing episode history, an integral part of medical health records and it also provides some challenges in time-changing data mining techniques. The application task was to examine time-changing features of medical treatment services for diabetics. This was approached by clustering patients into groups receiving similar patterns of care and visualising the features devised to highlight interesting patterns of care.

**Keywords:** time-changing data mining, event sequence, time-changing feature selection, hierarchical distribution, health records.

## 1 Introduction

A huge amount of data is collected every minute in the form of event time-changing sequences. Temporal data mining is concerned with discovering time-changing knowledge from these time-changing sequences. But one of basic problems of time-changing data mining is selecting useful and sufficient features for mining time-changing knowledge. Although a lot of work has been done on discovering time-changing patterns such as periodic patterns and similar patterns in discrete-valued time series (DTS) datasets(e.g. [2], [13]), little attention has been paid to the discovery of time-changing patterns or relationships that involve time-changing feature selection. We believe that time-changing feature selection is an important aspect of time-changing data mining.

In this paper we describe a new framework of time-changing feature selection for discovering patterns from time-changing health records. Time-changing feature selection (also known as time-changing attribute selection) is important to time-changing data mining because each feature component of a time-changing observation is based on

---

<sup>1</sup> Feature is also called in other names such as attribute, property and characteristic

time-changing measurements. The goal of time-changing feature selection is to make the error distribution on data mining results of time-changing behaviour (e.g., time-changing pattern, time-changing rules, time-changing cluster and so on.) as small as possible. In fact, the feature space of a large set of time-changing records is large and sparse, making it difficult for time-changing data mining to build good temporal data models. For example time-changing noise (e.g., noise with uncertainty time component) is one of the important problems in time-changing feature selection which makes meaningful time-changing clustering (or, classification) difficult.

The paper is organised as follows. Section 2 is devoted to the discussion of our framework based on hierarchical distribution for time-changing feature selection. In section 3 we first explain briefly the background of the application of our method and then describe how our selection methods are applied to time-changing health records and discuss the results of our experiments. Section 4 discusses related work and concludes the paper with a brief summary of our contributions.

## 2 Hierarchical Time-changing Feature Selection

In this section, we present our hierarchical time-changing feature selection method in searching and analysing time-changing features for the purpose of **Temporal Data Mining**.

For an analysis of a real-world time-changing sequence which may contain different kinds and levels of time-changing features such as complete and partial similarity time-changing features and periodicity time-changing features, we consider two groupings of the time-changing sequence,

1. Qualitative-based feature grouping, and
2. Quantitative-based feature grouping.

Then we combine the results from the above two groupings in a hierarchical fashion to obtain the final global time-changing features for time-changing data mining.

### 2.1 Some Definitions

We first give definitions for what we mean by time-changing feature, then provide some definitions and notations which will be used later.

**Definition 1** Suppose  $O_b = \{f_1, f_2, \dots, f_d\}$  is an observation with features  $f_j$  ( $1 \leq j \leq d$ ), if all or some of the features  $f_j$  always vary with time, then feature(s)  $f_j$  are called time-changing feature(s). Time-changing feature set  $T_f$  of a dataset consists of all features that vary with time.

**Definition 2** Hierarchical time-changing feature selection is a process that chooses an optimal time-changing feature subset according to its time-changing feature selection from its qualitative-based feature grouping and quantitative-based feature grouping (a certain criterion).

Our time-changing sequence analysis method is based on a time series measuring method that represents measurements of similarity (or, dissimilarity) among (or after transformation) the time series data.

**Definition 3** Suppose we have multiple and/or multidimensional time-changing series dataset such as

$$T = \left\{ \begin{pmatrix} \mathbf{t}_1 \\ \mathbf{x}_1 \end{pmatrix}, \begin{pmatrix} \mathbf{t}_2 \\ \mathbf{x}_2 \end{pmatrix}, \begin{pmatrix} \mathbf{t}_3 \\ \mathbf{x}_3 \end{pmatrix}, \dots, \begin{pmatrix} \mathbf{t}_n \\ \mathbf{x}_n \end{pmatrix}, \dots \right\}$$

where  $\mathbf{t}_i$  is the time vector component of an observation and  $\mathbf{x}_i$  is the value vector component of the observation, then we define the multiple and/or multidimensional **time-gap time series**  $T_g$ <sup>2</sup> of  $T$  as follows:

$$T_g = \left\{ \begin{pmatrix} \mathbf{t}_1 - a_1 \\ \mathbf{x}_1 \end{pmatrix}, \begin{pmatrix} \mathbf{t}_2 - a_2 \\ \mathbf{x}_2 \end{pmatrix}, \begin{pmatrix} \mathbf{t}_3 - a_3 \\ \mathbf{x}_3 \end{pmatrix}, \dots, \begin{pmatrix} \mathbf{t}_n - a_n \\ \mathbf{x}_n \end{pmatrix}, \dots \right\}$$

where  $\{a_1, a_2, \dots, a_n\}$  are values of a function  $f(t)$ <sup>3</sup>

For every successive three time points:  $t_j, t_{j+1}$  and  $t_{j+2}$ , the triple time-changing value of  $\{x_{t_j}, x_{t_{j+1}}, x_{t_{j+2}}\}$  has only nine distinct states. That is, let  $S_s$  be the same state as prior one,  $S_u$  the go-up (or, stronger) state compared with prior one and  $S_d$  the go-down (or, weaker) state compared with prior one. Then we have the following definition of a state-space  $S$ :

**Definition 4** Let  $S = \{s1, s2, s3, s4, s5, s6, s7, s8, s9\} = \{(x_j, S_u, S_u), (x_j, S_u, S_s), (x_j, S_u, S_d), (x_j, S_s, S_u), (x_j, S_s, S_s), (x_j, S_s, S_d), (x_j, S_d, S_u), (x_j, S_d, S_s), (x_j, S_d, S_d)\}$ , then  $S$  called state-space. If the triple time-changing value of  $\{x_{t_j}, x_{t_{j+1}}, x_{t_{j+2}}\}$  are all independent to each other, then  $S = \{1, 2, \dots, N\}$  is also called a state space.

The meaning of *Data* is a result of something exhibiting certain regularities, something representing a concept of what was observed. In general, *Data* is a triple such as

$$\text{Data} = \{v, \varrho, \rho\}$$

where  $v$  represents the quantitative set of the observation that can, at least in principle, be executed by some technical apparatus and  $\varrho$  is the qualitative set of the observation and  $\rho$  represents the position set of the observation<sup>4</sup>.

Here  $\rho$  is regarded as a monotonically increasing sequence of natural numbers. Each position in  $\rho$  is a time index for the corresponding values in  $v$  and  $\varrho$ .

**Definition 5** If  $\mathcal{V} = \{v, \rho\}$  is called a **quantitative set**. If  $\mathcal{Q} = \{\varrho, \rho\}$  is called a **qualitative set**.

<sup>2</sup> Sometimes, **time-gap time series**  $T_g$  is episode of an time-changing event

<sup>3</sup> In most cases we choose  $\mathbf{t}_i - a_i$  as the Euclidean distance between time component of the same behaviour between observations (e.g.,  $\mathbf{t}_i - \mathbf{t}_j$  or  $\mathbf{t}_i - \mathbf{t}_{i+1}$ ).

<sup>4</sup> In fact,  $v, \varrho$  and  $\rho$  are vectors.

**Example** We now present an example to illustrate the concepts introduced above. Suppose a dataset  $\mathcal{D}_t$  consists of 3 months of daily U.S. dollar exchange rate against Canadian dollar (e.g., 90 points):

$$1.318, 1.3215, 1.3235, 1.3181, \dots, 1.3534, 1.3561, 1.3575, 1.3569, 1.3573$$

It can be transformed into state-space  $\mathcal{S}$  as a qualitative sequence of the data (here we write  $i$  for  $s_i \in \mathcal{S}$ ):

$$\mathcal{Q} = \{1, 3, \dots, 1, 1, 3, 9, 7\} \quad (1)$$

Then we have  $\mathcal{D}_t = \{d_1, d_2, \dots, d_m\}$  in the form of both quantitative and qualitative<sup>5</sup>:

$$\mathcal{D}_t = \left\{ \begin{pmatrix} 1.318 \\ 1.3215 \\ 1.3235 \end{pmatrix} \times 1, \begin{pmatrix} 1.3215 \\ 1.3235 \\ 1.3181 \end{pmatrix} \times 3, \dots, \begin{pmatrix} 1.3561 \\ 1.3575 \\ 1.3569 \end{pmatrix} \times 9, \begin{pmatrix} 1.3575 \\ 1.3569 \\ 1.3573 \end{pmatrix} \times 7 \right\}$$

We can also have  $\mathcal{D}_t = \{d_1, d_2, \dots, d_m\}$  in a natural way to form the sequence in both of the quantitative and qualitative forms:

$$\mathcal{D}_t = \{(1.318) \times 1, (1.3215) \times 3, \dots, (1.3561) \times 9, (1.3575) \times 7\}$$

This is a principal representation of the sequence in both forms of quantitative and qualitative. It is clear that for any dataset there exist only one quantitative set and only one qualitative set. In other words the resolution of any dataset into its quantitative set and its qualitative set is unique. We will use this form to develop a time-changing feature selection method for data mining purposes in later sections.

## 2.2 Hierarchical Feature Selection

We assume that for each successive pair of time points in a DTS, we have  $t_{i+1} - t_i = f(c)$  (a uniformly distributed function). We consider the bivariate data  $(X_{t_1}, Y_{t_1}), \dots, (X_{t_n}, Y_{t_n})$ , which forms an independent and identically distributed sample from a population  $(\mathbf{X}, \mathbf{Y})$ . For given pairs of data  $(X_{t_i}, Y_{t_i})$ ,  $i = 1, 2, \dots, N$ , we can regard the data as being generated from the model

$$\mathbf{Y}_{t_i} = m(\mathbf{X}_{t_i}) + \sigma(\mathbf{X}_{t_i})\varepsilon \quad (2)$$

where  $\mathbf{E}(\varepsilon) = 0$ ,  $\mathbf{Var}(\varepsilon) = 1$ , and  $\mathbf{X}_{t_i}$  and  $\varepsilon$  are independent<sup>6</sup>.

In other words, the data model also corresponds to its submodels, which are called **quantitative data model** and **qualitative data model**, such as:

$$\mathcal{V}_{t_i} = m(v_{t_i}) + \sigma(v_{t_i})\varepsilon \quad (3)$$

$$\mathcal{Q}_{t_i} = m(\varrho_{t_i}) + \sigma(\varrho_{t_i})\varepsilon \quad (4)$$

<sup>5</sup> Where the symbol " $\times$ " represents a relationship between the data quantitative and data qualitative.

<sup>6</sup> We always denote the conditional variance of  $\mathbf{Y}$  given  $\mathbf{X} = \mathbf{x}_0$  by  $\sigma^2(\mathbf{x}_0)$  and the density of  $\mathbf{X}$  by  $f(\bullet)$ .

**Qualitative Time-changing Feature Selection** Qualitative time-changing feature selection is based on finding time-changing features from **data qualitative set**, and the qualitative data set is based on state-space  $\mathcal{S}$ . We first suppose that a qualitative sequence on  $\mathcal{S}$  as a set of structural vector sequences, such as  $\{\mathbf{S}_1, \dots, \mathbf{S}_m\}$ , where each  $\mathbf{S}_i = (s_{i1}, s_{i2}, \dots, s_{in})^T$  denotes the  $n$ -dimensional time-changing attributes for each time-changing object  $\mathbf{S}_i$  that is to be assigned to a prespecified distribution class.

Let  $\{\mathcal{Q}_t : t \in \mathbf{N}\}$ , where  $\mathcal{Q}_t = \{\varrho_1, \varrho_2, \dots, \varrho_k\}$  is the data qualitative sequence and (for  $1 \leq j \leq k$ )  $\varrho_j \in \mathcal{S} = \{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9\}$ , be an irreducible homogeneous qualitative sequence<sup>7</sup> on  $\mathcal{S}$ , with probability sequence  $\Gamma = (\gamma_{ij})$ , where for all qualitative states  $s_i$  and  $s_j$  and times  $t$ :

$$\gamma_{ij} = \mathbf{P}(\mathcal{Q}_t = s_{ij})$$

Also we define a probability sequence which is a correlated measure of the relationship of two time-changing features. It is called a **correlation ratio sequence** for all qualitative states  $s_i$  and  $s_j$  and times  $t$ :

$$\varpi_{ij} = \mathbf{P}\left(\frac{\mathcal{Q}_t(s_j)}{\mathcal{Q}_t(s_i)}\right).$$

For each  $\mathcal{Q}_t$ , there exists a unique, strictly positive, statistical distribution.

**Quantitative Time-changing Feature Selection** On the quantitative time-changing feature selection, we consider the relationship between the response time-changing feature variable  $\mathcal{V}_t$  and the vector of time-changing feature variables  $v = (t, v_1, \dots, v_n)^T$ . For a given dataset of generated data model 2, the unknown regression function  $m(\mathbf{x})$  is obtained by applying a Taylor expansion of order  $p$  in a neighbourhood of  $\mathbf{x}_0$  with its remainder  $\vartheta_p$ ,

$$m(\mathbf{x}) = \sum_{j=0}^p \frac{m^{(j)}(\mathbf{x}_0)}{j!} (\mathbf{x} - \mathbf{x}_0)^j + \vartheta_p \equiv \sum_{j=0}^p \beta_j (\mathbf{x} - \mathbf{x}_0)^j + \vartheta_p.$$

The first stage of methods for detecting the characteristics of those records is to use the linear regression analysis. We may assume linear model  $\mathbf{Y} = \mathbf{X}\beta + \varepsilon$ . The linear model based upon least square estimation (LSE) is  $\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$ . Then we have:  $\hat{\beta} \sim N(\beta, \text{Cov}(\hat{\beta}))$ . Particularly, for  $\hat{\beta}_i$  we have  $\hat{\beta}_i \sim N(\beta_i, \sigma_i^2)$ , where  $\sigma_i^2 = \sigma^2 a_{ii}$ , and  $a_{ii}$  is the  $i$ th diagonal element of  $(\mathbf{X}^T \mathbf{X})^{-1}$ .

Now, for the set of pure values, we may fit a local linear model as above and parameters can be estimated under  $\mathcal{LSE}$ . Then the problem can be formulated as the data distribution functional analysis of discrete-valued time series.

**Global Time-changing Feature Selection** We combine the above two kinds of feature discovery to discover global time-changing features from a time-changing dataset. In the qualitative group, let the qualitative sequence  $\{\mathcal{Q}_t : t \in \mathbf{N}\}$  be data functional

<sup>7</sup>  $s(i+1)$  only depends on  $s(i)$ .

distribution sequence on the state-space  $\mathcal{S} = \{s1, s2, s3, s4, s5, s6, s7, s8, s9\}$ . Then suppose the data quantitative sequence is a nonnegative random vector process  $\{\mathcal{V}_t; t \in \mathbf{N}\}$  such that, conditional on  $\mathcal{S}^{(T)} = \{\mathcal{Q}_t : t = 1, \dots, T\}$ , the random vector variables  $\{\mathcal{V}_t : t = 1, \dots, T\}$  are mutually independent. We give an example later to show how to mine global time-changing features from a dataset.

Suppose that, if  $\mathcal{Q}_t = si$ ,  $\mathbf{Y}_t$  has a Poisson distribution with mean  $\lambda_i$ , let  $E(\mathbf{Y}_t | \mathcal{Q}_t)$ , the conditional mean of  $\mathbf{Y}_t$  be

$$\mu(t) = \sum_{i=1}^m \lambda_i W_i(t),$$

where the random variable  $W_i(t)$  is the indicator of the event  $\{\mathcal{Q}_t = si\}$ . The state-dependent probabilities are then given for all nonnegative  $v$  by

$$\pi_{vi} = \frac{e^{-\lambda_i} \lambda_i^v}{v!}$$

The models  $\{\mathbf{Y}_t\}$  are defined as Poisson hidden models.

### 3 An Application in Medical Records

In this section, we first explain briefly the background of our application and then we present our time-changing feature selection techniques on the analysis of the medical service profiles of diabetes, a common disease in the senior population in Australia. We have applied our technique to the dataset and identified the distribution of time-changing features of the diabetes patients.

#### 3.1 The Background of the Application

Medicare is the Australian Government's universal health care system. Each visit to a medical practitioner or hospital is covered by Medicare and recorded as a transaction in the Medicare Benefits Scheme (MBS) database. This data has been collected in Australia since the inception of Medicare in 1975. Such a massive collection of data provides an extremely rich resource that has not been fully utilised in the exploration of health care delivery in Australia. The HIC<sup>8</sup>, has a responsibility to protect the public purse and to ensure that taxpayer's funds are spent wisely and efficiently on health care. The knowledge discovered can be used to educate medical practitioners to improve their medical practice in order to achieve the best health outcomes while ensuring health costs remain under control. For this current exploration we use a subset of de-identified data (to protect privacy) based on Medicare transactions for the period 1997 to 1998. Our particular focus is on time-changing feature selection for data mining related to care models for diabetes. For example, we can ask ourselves questions like: Are there any distinct time-changing features of care for these diabetes patients? Are there any groups

<sup>8</sup> The Health Insurance Commission of Australia: <http://www.hic.gov.au>

of patients receiving similar time-changing features of care? Are the time-changing features of care related to their doctor? Do patients of different ages or gender or location receive differing time-changing features of care to other patients? Answers to the above questions rely on a thorough analysis of the sequences of medical test of the patients and is the objective of our application.

In particular, the purpose of this experiment is to find a set of time-changing features for describing the patterns of care in the management of diabetes. These time-changing features will provide input to a model that will monitor behaviour by diabetic patients. The data used in this case study was extracted from the Medicare transactional database<sup>9</sup>. The data extracted from Medicare is raw transaction data which is stored on IBM main 370 frame computer running MVS operating system. It is a very large data set with millions of records and each record has more than a hundred attributes. There is a transaction record for each Medicare service. Each service record has its item number which is the most important field in the data set. The item number tells to a large extent what kind of service has been performed on the patient. The patients' medical service pattern is represented by a series of item numbers served during the year. The records include fields such as: Encrypted Provider number, Encrypted PIN number, Method of Payment, Item number, Date of service (DOS), Benefit, Reason code for rejection, Referral provider, Processing indicator, Date of referral and Hospital index. From each patients' medical service pattern, an **EPISODE**<sup>10</sup> dataset is generated:

Patient, DOS, Item1, Item2, refs, Item3, Item4, . . .

where the first item(s) have no referral date and the remaining items have a referral date same as the first item(s) DOS. In most cases, the episode is not the same for each time period. For example, there is a patient, whose episode sequence in the six Medical categories<sup>11</sup> (e.g., 1 is stand for category one) benefits schedule is:

```
1, 6, 6, 6, 1, 1
1
2, 1, 5, 1, 6, 6, 6, 1, 1, 3, 5, 5
1
2, 6, 6, 6, 2, 5, 1, 1,
1, 6, 6, 6, 6, 1, 6, 6, 1, 1, 2
```

From each medical category, the episode includes all medical items that have been used and associated with doctors, DOS and so on.

<sup>9</sup> The data consist of 10,000 diabetic patients using Medicare services paid by H.I.C. under the Medicare Benefits Schedule.

<sup>10</sup> The EPISODE is time period. The time a patient spends in the continuous care of consultants using Medicare services which paid by H.I.C. of one provider (e.g. GP) or, in the case of shared care, in the care of two or more consultants. Where care is provided by two or more consultants within the same episode, one consultant will take overriding responsibility for the patient and only one consultation EPISODE is recorded.

<sup>11</sup> The six categories of medical services are: professional Attendances (PA); Diagnostic Services (DS); Approved Dental Practitioner Services (AD); Diagnostic Imaging Services (DI); and Pathology Services (PS).

In this experiment, we use our new method to analyze the medical service profiles of diabetes. We have applied our technique to identify classes, in which the patterns of the diabetic patients were found. The event sequence data can be augmented with any available vector based data. There are three steps of time-changing feature selection from the time-changing sequences (databases):

1. Data quantitative time-changing feature selection,
2. Data qualitative time-changing feature selection, and
3. Data global time-changing feature selection by the above two steps.

Through this experiment, for example, we are interested in finding following time-changing features:

- Does there exist any global time-changing feature  $T_{f_t}$  for doctor visits by all diabetic patients?
- What kind of sub-time-changing features exist for all diabetic patients?
- What kinds of time-changing features (models) are there for groups of diabetic patients?

### 3.2 Steps for the Application of Hierarchical Feature Selection

The main steps for applying **Hierarchical Feature Selection** within time-changing data mining are as follows:

- The formalisation of the time-changing data mining problem for hierarchical time-changing feature selection,
- Selecting two different hierarchical time-changing features from qualitative and quantitative parts, then selecting a subset of time-changing features for global time-changing feature set, and
- The interpretation of the global time-changing feature informations.

### 3.3 On Qualitative Time-changing Feature Selection

The first experiment is the selection of qualitative time-changing feature. We are investigating the data qualitative time-changing feature on state-space  $\mathcal{S} = \{s1, s2, s3, s4, s5, s6\}$ <sup>12</sup> to test the naturalness of qualitative time-changing features. For selecting all levels of time-changing features, we use time gap functions with a distance function on state-space  $\mathcal{S}$ .

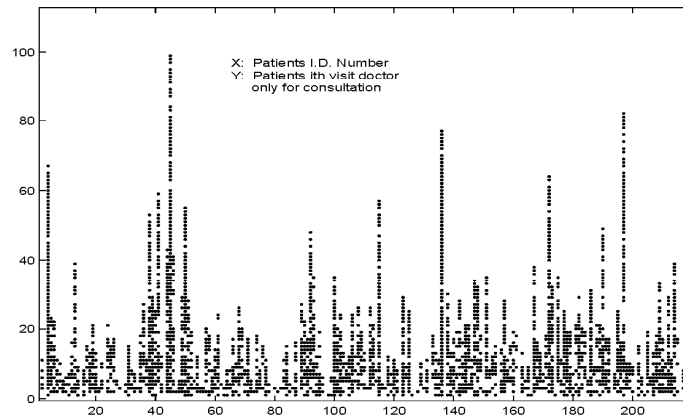
In this qualitative time-changing feature selection experiment, all items have been grouping into six categories for each patient. To find the best subset of time-changing features, the key issue is that features between categories that are highly correlated or have mutual information will have similar weight values in categories. The selection process can be summarized in the following steps:

<sup>12</sup> There are only six categories in MBS Australia

- step 1** Grouping episode sequence for each patient into six categories by its weighted probability values. Computing the sample correlation matrix for those six categories.
- step 2** Computing standard deviation on **correlation ratio sequence** to find degree of relationship between categories then grouping them into different feature classes.
- step 3** Clustering six categories according to above feature classes into sub-feature classes within each of six categories according to time gap distribution function.
- step 4** Choose qualitative time-changing features from those relationship classes. Those features can be represented by a path between categories.

Now we interpret three important (and smallest) subsets of time-changing features we have discovered from the qualitative time-changing sequences:

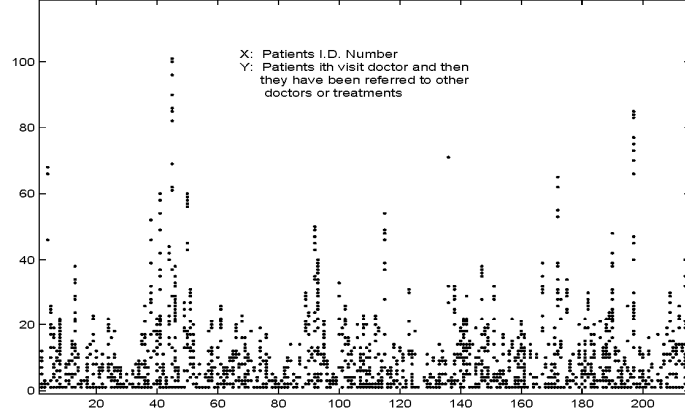
- There exists a moderate time-changing feature (similarity) relationship between MBS category one and MBS category six for all diabetic patients visiting their doctors regardless of whether they have followed up their visits by any medical treatments (e.g., between Figure 1 and Figure 2).
- For non-diabetic medical treatments of diabetic patients, there does not exist any correlated time-changing feature between any of the categories. This means, for example, there are non time-changing related common features among diabetic patients other than problems related to the diabetes.



**Fig. 1.** Time-changing qualitative feature distribution: patients only use consultation item numbers.

### 3.4 On Quantitative Time-changing Feature Selection

We now illustrate our new method to analyse the quantitative sequence of health time-changing records for selecting time-changing features. In this health time-changing



**Fig. 2.** Time-changing qualitative feature distribution: patients use consultation and medical item numbers.

records, since each patient record length is different, we can only use their statistical value as variables in regression functions. In the light of our selected qualitative time-changing feature in first the experiment, we have the series

$$Y_t = f_t^{feature_i}(v_t) - f_t^{feature_j}(v_t)$$

where  $f_t^{feature_i}(v_t)$  is a frequency distribution function of medical item numbers which have been used within **feature i**, its variable  $v_t$  is the time distance between the same feature (e.g.,  $v_t = feature_{k_{t1}} - feature_{k_{t2}}$ ), in the same time-changing cluster. Then the observations can be modelled as a linear regression function, say

$$Y_t = f_t^{feature_i}(v_t) - f_t^{feature_j}(v_t) + \varepsilon_t, \quad t = 1, 2, \dots, N$$

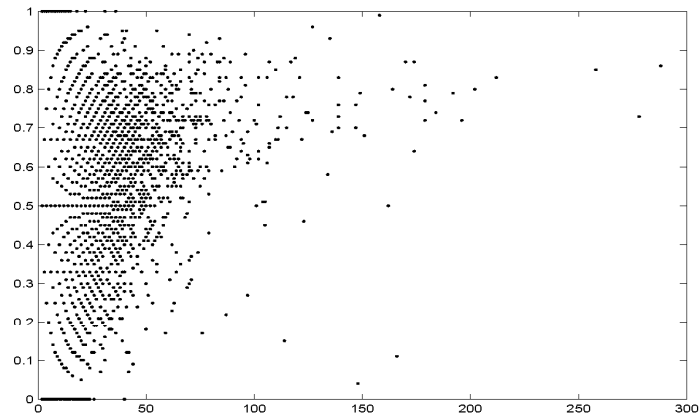
and we also consider the  $\varepsilon(t)$  as an auto-regression  $AR(2)$  model

$$\varepsilon_{t'} = a\varepsilon_{t'-1} + b\varepsilon_{t'-2} + e_{t'}$$

where  $a, b$  are constants dependent on sample dataset, and  $e_{t'}$  with a small variance constant which can be used to improve the predictive equation.

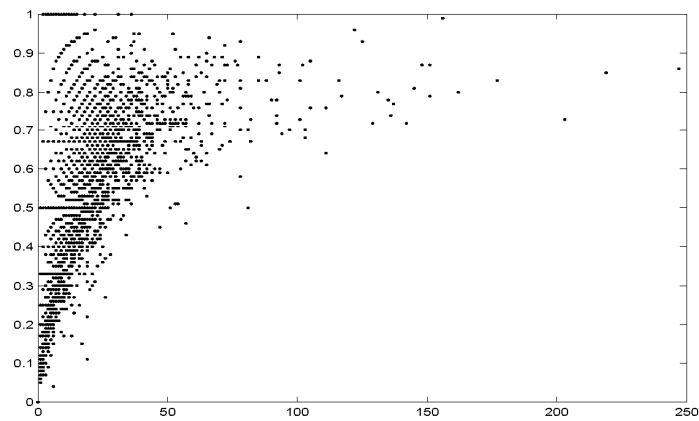
For each of the time-changing qualitative features found there exist two important quantitative time-changing features. Some results of quantitative time-changing features experiments are explained as follows.

In Figure 3, the x-axis represents the number of each item number that has been used in different time-changing qualitative features and the y-axis represents the probability of the item within the time-changing cluster  $k$ . This explains two facts: (1) the distribution of the number of items the patients have received is periodic and similar for all the medical treatments related to all kinds of diabetic problems. And (2) that patients have received treatment according to the medical guidelines.



**Fig. 3.** Item frequency distribution in the different time-changing features.

But in Figure 4, the  $x$ -axis represents the number of each item number that has been used in the same selected time-changing qualitative feature over all medical categories and the  $y$ -axis represents the probability of the item within the selected time-changing qualitative feature. This also explains two facts: for example, (1) the patients have received moderately similar medical treatments for diabetic problems, because doctors have different levels of knowledge of diabetes problems. And (2) that patients have received a number of medical treatment items that relate to diabetes problems (e.g., the heart problems and eye problems) depending on their doctors knowledge.



**Fig. 4.** Item frequency distribution in the same time-changing qualitative feature.

### 3.5 Time-changing Feature Selection for Global Mining

According to the results from qualitative time-changing feature selection and quantitative time-changing feature selection in the health data records, let  $\{S_t : S_t \in \mathcal{S}, t \in \mathbf{N}\}$  be a qualitative process representing *features* occurrence, and  $\{V_t : t \in \mathbf{N}\}$  be the corresponding to qualitative process which includes number of all item's medical number used sequence, then we have the global features of  $V_t$  conditional distribution on  $S_t$  given by

$$\mathbf{P}(V_t = v | S_t = i) = p_{vi}^t$$

Then some results of global feature experiments can be explained as follows.

- **feature 1:** In state space  $\mathcal{S}$ ,  $V_t^{state1}$  and  $V_t^{state6}$  both have Poisson distribution with means  $\lambda_i^{state1}$  and  $\lambda_i^{state5}$ . Two states satisfy the condition  $V_t^{state1} = \alpha V_t^{state5} + \theta_t$ . Then the conditional mean of  $V_t$  and state-dependent probabilities given for all non-negative integers  $v_t$  will be

$$\mu(t) = \sum_{i=1}^m \lambda_i W_t(t), \quad P_{v_t, statek} = e^{-\lambda_{i,v_t}} \frac{\lambda_{i,v_t}^{v_t}}{v_t!}$$

- **feature 2:** For state 2,  $V_t^{state2}$  is an exponential distribution with parameters  $\lambda_i^{state2}$  and  $\mu_i^{state2}$ . Then the conditional exponential distribution of  $V_t^{state2}$  and state-dependent probabilities given for all non-negative integers  $v_t$  will be

$$m(t) = \sum_{i=1}^m \lambda_i W_t(t), \quad P_{v_t, test2} = \begin{cases} \lambda_{(i,v_t)} e^{(-\lambda_{(i,v_t)})(v_t - \mu)} & v_t > \mu \\ 0 & v_t < \mu \end{cases}$$

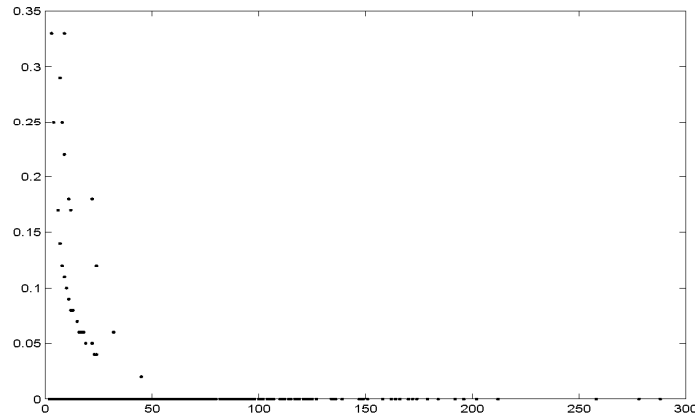


Fig. 5. Item frequency distribution in the time-changing qualitative feature 2

- **feature 3:** For state 3,  $V_t^{state3}$  is a geometric distribution with parameter  $p_i^{state3}$ . Then the conditional geometric distribution of  $V_t^{state3}$  and state-dependent probabilities given for all non-negative integers  $v_t$  will be

$$m(t) = \sum_{i=1}^m p_i W_t(t), \quad P_{v_t, test3} = p_{i, v_t}^{state3} (1 - p_{i, v_t}^{state3})^{(v_t-1)}.$$

- **other feature:** States 4 and 5 are independent. We found that there exist some similarity patterns between their states but non-related time-changing features exist between their clusters, this means the patients have received a number of treatments of time-changing features that are similar but for different time periods.

According to those important time-changing features, we can start mining the health data records. The main result from qualitative pattern search and quantitative pattern search is that the behaviour of doctor visits toward a pure diabetic (e.g., the patient has only one medical diabetic problem) is a Poisson distribution. The other main combined-results of time-changing data mining on the health dataset are as follows:

- According to time-changing feature 1, there exist some full periodic time-changing patterns among the diabetes patients, especially the patterns across over (MBS) category 1 and category 6.
- There exist some partial similarity patterns if time-changing feature 2 has been used over all MBS categories. This means that the patients have sub-common problems (for instance, they all have an eye problem) and the distribution of uncommon problems is non-stationary, etc.
- There does not exist any full periodic pattern of medical test item number for all diabetes's patients, but there exist some similarity patterns with a small time gap shift for those item numbers.

## 4 Discussion and Conclusion

In recent years, various studies have been done in knowledge discovery from time-changing datasets for searching different kinds of and/or different levels of patterns, but the techniques are not for general cases. For example, most researchers use statistical techniques such as Metric-distance based technique, Model-based technique, or a combination of techniques (e.g., [6], [14]) to search different pattern problems such as in periodic pattern searching, e.g., [5, 7], in similarity pattern searching (e.g., [3]). In [1], R. Agrawal and others present a “shape definition language”, called *SDL*, for retrieving objects based on shapes contained in the histories associated with these objects. In [4], Das with others describe adaptive methods which are based on similar methods for finding rules and discovering local patterns and in [12] Rohan and others have considered three alternative feature vectors for representing variable-length patient health records.

Our work is different from their work. First, we use a statistical language to perform all the search work based on time-changing features identified from the dataset. Second, we divide the data sequence or, data vector sequence into two groups for time-changing feature selection: data quantitative group and data qualitative group.

We have considered the use of data feature vectors of quantitative and qualitative groups for representing variable-length patient health records. The time-changing feature of the qualitative group is the simplest one found, but important since it does capture the distribution of patient care throughout the data window. For the time-changing data mining task for discovering diabetes care relationship patterns, the time gap feature within both quantitative group and qualitative group most directly represents in selected global time-changing features. We expect that our **Hierarchical Time-changing Feature Selection** method presented here for event sequence data for this health application will be applicable to other time-changing event sequence data such as market trading dataset.

This paper has presented a new approach based on hierarchical time-changing feature selection of application of time-changing data mining. The clusters of similarity patterns are computed in this level by the choice of certain time gap measures. The quantitative patterns are decided in the second level and the similarity and periodicity of a DTS are extracted. In the final level, we combine qualitative and quantitative features to obtain a global pattern picture and understand the patterns in a dataset better. Another approach to find similar and periodic patterns has been reported in [8–11]; there the models used are based on hidden functional analysis. The use of time-changing feature selection in our framework makes mining time-changing patterns from time-changing health records easier and faster. Most importantly, it increases the quality of the discovered patterns.

## References

1. Rakesh Agrawal, Giuseppe Psaila, Edward L. Wimmers, and Mohamed Zait. Querying shapes of histories. In *Proceedings of the 21st VLDB Conference*, September 1995.
2. C. Bettini. Mining temporal relationships with multiple granularities in time sequences. *IEEE Transactions on Data & Knowledge Engineering*, 1998.
3. G. Das, D. Gunopulos, and H. Mannila. Finding similar time series. In *Principles of Knowledge Discovery and Data Mining '97*, 1997.
4. G. Das, K. Lin, H. Mannila, G. Renganathan, and P. Smyth. Rule discovery from time series. In *Proceedings of the international conference on KDD and Data Mining (KDD-98)*, 1998.
5. J. Elder IV and D. Pregibon. A statistical perspective on knowledge discovery in databases. In U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 83–115. The MIT Press, 1995.
6. J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.
7. Cen Li and Gautam Biswas. Temporal pattern generation using hidden markov model based unsupervised classification. In *Proc. of IDA-99*, pages 245–256, 1999.
8. Wei Q. Lin and Mehmet A. Orgun. Applied hidden periodicity analysis for mining discrete-valued time series. In *Proceedings of ISLIP-99*, pages 56–68, Demokritos Institute, Athens, Greece, 1999.
9. Wei Q. Lin and Mehmet A. Orgun. Temporal data mining using hidden periodicity analysis. In *Proceedings of ISMIS-2000*, University of North Carolina, USA, 2000.
10. Wei Q. Lin, Mehmet A. Orgun, and Graham Williams. Temporal data mining using multilevel-polynomial models. In *Proceedings of IDEAL-2000*, The Chinese University of Hong Kong, Hong Kong, 2000.

11. Wei Q. Lin, Mehmet A.Orgun, and Graham Williams. Temporal data mining using local polynomial-hidden markov models. In *Proceedings of PAKDD-2001*, The University of Hongkong, Hong Kong, 2001.
12. G. Williams R. Baxter and H. He. Feature selection for temporal health records. In *The Fifth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-01)*, Hong Kong, April 16-18, 2001. Springer-Verlag.
13. John F. Roddick and M. Spiliopoulou. A survey of temporal knowledge discovery paradigms and methods. *IEEE Transactions on Knowledge and Data Engineering*, 13, 2001.
14. Z.Huang. Clustering large data set with mixed numeric and categorical values. In *1st Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 1997.



# A Multi-level Framework for the Analysis of Sequential Data

Carl H. Mooney, Denise de Vries and John F. Roddick

School of Informatics and Engineering  
Flinders University of South Australia,  
PO Box 2100, Adelaide, South Australia 5001,  
{carl.mooney, denise.devries, roddick}@infoeng.flinders.edu.au

**Abstract.** Traditionally text mining has had a strong link with information retrieval and classification, for search engine purposes, and has aimed to classify documents according to known knowledge. Association rule mining and sequence mining on the other hand have had a different goal; one of eliciting relationships within or about the data being mined. Recently there has been some research conducted using sequence mining techniques on digital document collections by treating the text as sequential data.

In this paper we propose a multi-level framework that is applicable to text analysis and that improves the knowledge discovery process by finding additional or hitherto unknown relationships within the data being mined. We believe that this can lead to the detection or fine tuning of the context of documents under consideration and may lead to a more informed classification of those documents. Moreover, since we use a semantic map at varying stages in the framework, we are able to impose a greater degree of focus and therefore a greater transitivity of semantic relatedness that facilitates the improvement in the knowledge discovery process.

## 1 Introduction

Association rule mining, sequence mining and more recently text mining, in common with other knowledge discovery algorithms, all have a similar goal, namely to extract useful knowledge from large amounts of data. The techniques employed by association and sequence mining have been analogous and aim to elicit relationships within or about the data. In contrast, text mining, which has a strong link to information retrieval and classification for search engine purposes, largely aims to classify documents according to known knowledge. There has, however, been some research conducted into the application of data mining techniques for text mining based on the assumption that text is sequential data [2, 3, 9] and it from this perspective that we intend to use our developed framework.

Data mining has used constraints placed on the mining process to accomplish, among other things, the reduction of the search space (support thresholds etc.),

the adjustment of what is relevant (item constraints), and by incorporating these traditional constraint heuristics, in particular support, some elements have been lost from consideration due to a failure to meet the designated support. To a certain extent the problem of what to do with lower level concepts (elements), that failed to meet a support threshold, was alleviated with the introduction of hierarchies that enabled the failed concepts to be viewed within an encompassing concept further up the tree. Inferences could then be drawn from both within each level or across each level of the hierarchy. However, a possible problem still existed; What if the reason for failure to meet support was due to a simple typographic error during the input of the data? For example, if the terms **hot** and **sit** narrowly fail to meet the support threshold because some occurrences have been entered as **sot** instead of **sit** or **hit** instead of **hot**, it follows that the inclusion of the correct word will be reflected in the support value, and thus the terms will meet threshold. This would necessarily give an improved, and truer, picture or representation, of the data. On the next level, that of similar concepts, for example Bush – President, there exists a similar problem that may also have an impact. There may be variations in correct spelling, for example Al-Qaida al-Qaeda, al-Qa'ida, al-Quaida, el-Qaida, äI-Qaida, al Qaeda or al Quaeda.

In medicine, there are often duplicate terms for the same illness or symptoms, dependent upon the branch of medicine, as well as the same or similar term for different conditions. For example, scrapie, ovine spongiform, bovine spongiform encephalopathy (BSE), Mad Cow Disease, kuru, Creutzfeldt-Jakob disease (CJD), new variant Creutzfeldt-Jakob disease (nvCJD), and transmissible mink encephalopathy (TME), all refer to the same disease in sheep, cattle and humans.

Of course we recognise that some of these semantic similarities are highly context sensitive and that the 'Bush – President' example would not be similar in a botanical context, but the ability to 'compress' these two concepts (i.e. process as one concept) for the purpose of mining<sup>1</sup> would necessarily maintain the intention of the data and perhaps produce otherwise unknown relationships, be they simple associations, or more complex temporal relationships.

The remainder of this paper is organised as follows. Section 2 presents related work in the areas that are encompassed by our framework, Section 3 discusses the framework that has been implemented thus far, and Section 4 concludes with some discussion of future work.

## 2 Related Work

### 2.1 Sequence Mining

Sequence mining is not limited to data stored in overtly temporal or longitudinally maintained datasets – examples include genome searching, web logs, alarm data in telecommunications networks, population health data etc. In such

<sup>1</sup> We acknowledge the need to calculate a relevant support that would be indicative of the combined terms

domains data can be viewed as a series of events occurring at specific times and therefore the mining problem becomes a search for collections of events (episodes) that occur frequently together. Solving this problem requires a different approach, as opposed to the more traditional market-basket domain, and several types of algorithm have been proposed for different domains. For example Manilla *et al.* [26,27] have developed algorithms and evaluated them on alarm detection data. There is however no reason why text cannot be viewed in the same way and Ahonen *et al.* [2,3] and Rajman and Besançon [9] have applied similar techniques, based on generalised episodes and episode rules, to text analysis tasks.

In a ‘normal’ sequence mining scenario based on generalised episodes, each of the tokens are generally independent of each other, e.g. a message from sensor *A* is independent of a message from sensor *B*, unless *B* is reacting as a consequence, and the mining process uses a *sliding window* to limit the length of the discovered episodes. On this level also there are no semantics associated with each input token. The following example will serve to illustrate this point.

*Example 1.*

Given the following series of sensor readouts:

*A C B D F A C D F ...*,

then there is no reason that a *sliding window* cannot partition this at every token, and in essence this is what actually happens, resulting in the sequences below being generated.

$$\begin{array}{llll} \langle A \rangle & \langle AC \rangle & \langle ACB \rangle & \langle ACBD \rangle \dots \\ \langle C \rangle & \langle CB \rangle & \langle CBD \rangle & \dots \\ \langle B \rangle & \langle BD \rangle & & \dots \end{array}$$

In a text environment, however, (text files, emails etc.) this does not necessarily make sense due to the semantics of the language. The following example will serve to illustrate this.

*Example 2.*

Given the following sequence:

*Fred goes shopping on Tuesday. His cat is black.*

it makes no sense if we use the traditional *sliding window* method, as is the case in Example 1, because at some point we would end up with a sequence like

*on Tuesday His cat is black*

Although this may or may not be grammatically correct in its current context, in the context of the sequence from which it came, it is not semantically correct. Furthermore this sequence may never be frequent, but may have to be processed anyway due the length of the *window* under consideration. This type of problem

will exist for all types of text documents and therefore there is a need to further constrain the window under consideration. In Section 3 a strategy will be outlined that deals with this specific problem.

One of the problems for any data mining task is how to handle the voluminous amounts of data to be mined. Different strategies have been employed to improve the efficiency of the process and in the context of text mining this is of paramount importance. There are obvious benefits to employ mappings of the data, especially when dealing with text. These can be categorised mainly by space and time benefits, but in some instances it may be that there is a logical benefit. Of course certain things are evident when dealing with numeric mappings when the number of elements is greater than nine, the least of which is that there has to be some sort of delimiter involved to disambiguate the numbers. Having said this, the trade-offs are still better than mining the straight text. One problem does arise when you wish to perform any sort of semantic distance metric during the mining process. This will necessitate the conversion back to text to perform the test and then a realignment of the mappings on completion of the process. Although this is not conceptually difficult it may be time consuming in practice and also may pose some problems when realigning.

In our current endeavours the possibilities of encoding the text, so as to more efficiently process the amount of data, is not feasible since we need to compare both typographical and semantic differences or similarities between episodes.

## 2.2 Approximate String Matching

Research in the area commonly known as “string matching”, or “string edit distance” has been ongoing for quite some time and includes not only algorithms for string matching using regular expressions [16, 1, 10, 8, 23, 44, 12, 4], but also algorithms in the related area of edit distance [49, 48, 11, 34, 35, 13, 5, 6, 14, 7, 18]<sup>2</sup>. In particular the work by Oommen *et al.* [34, 35] has resulted in what has been called a *confusion matrix*. This matrix is for determining the probability of striking the wrong keys on a keyboard and then incorporating this into the edit distance function. The implementation of this and the incorporation of it into our framework has been accomplished and can be used during a first pass of the data, or as a pre-processing step in cleaning the data. It is envisaged that this will eliminate a considerable amount of typographic errors, but the problem of similar context sensitive concepts, e.g. Bush – President could also have an impact, therefore a level of processing to handle *semantic distance* is included in our framework.

## 2.3 Semantic Distance

There are many research areas, other than Data Mining and Knowledge Discovery, dealing with Semantic Distance: Knowledge Representation, Statistical Clus-

<sup>2</sup> For an excellent survey on this field see “A Guided Tour to Approximate String Matching” by Gonzalo Navarro [32]

tering, Machine Learning, Medical Informatics, and Natural Language Processing to name but a few. Semantic distance, in text, is a measure of the relationship between the *concepts* represented by words. How closely related they are depends upon their formal definition, their common usage and human psychology (where one term prompts us to think of the other). A thorough review of work done in these areas is beyond the scope of this paper, however, much of the work related to semantic distance relies on the linguistic or semantic similarity of terms that are based on a lexicographic definition of words or terms without reference to the application context. Approaches to measuring semantic similarity fall into three main types - thesaurus, dictionary and ontology based.

Thesaurus-based approaches use groups of related words and synonyms, usually arranged as a taxonomy, to determine similarity which is expressed generally a boolean value, *close* or *not close*. Synonymy judgement and semantic similarity research on pairs of words done by Rubenstein and Goodenough [43], Miller and Charles [28], Morris and Hirst [31], Okumura and Honda [33] provide methods to calculate the degree of similarity as a number. A limited 'transitivity' measure was also introduced by Morris and Hirst to provide a metric to trace patterns of lexical cohesion in text.

Dictionary-based metrics using hierarchical schemata evaluate the similarity of terms based on spatial or mereological classifications (that is, part-whole relationships) of entities or by hypernyms and hyponyms (specialisation-generalisation relationships) by edge counting methods in which the granularity of the description of entities and their classes affects the value of the distance.

Kedad and Métais [20] propose using metadata, including a linguistic dictionary, in a hierarchical structure with no distance set by the user. In their model, values are considered close if they belong to the same class. The semantic distance is fixed by the dictionary definition. The semantic similarity dependent upon a particular context is not readily extracted. Kozima and Furugori [21, 22] present a method which computes the semantic similarity between words using a semantic network constructed from a subset of the Longman Dictionary of Contemporary English. This method relies on each word used in a definition of a word being present in the dictionary.

A much-used semantic network resource is *WordNet* [15] developed at Princeton by George Miller. WordNet is comprised of separate networks for nouns, verbs, adjectives and adverbs with the basic element being a set of synonyms (synset). These can be arranged in *IS-A* hierarchies of hyponyms/hypernyms or with the additional relationships of meronyms (part-of), holonyms (has-part) and antonyms (opposite) as a multi-relational network. Many similarity measures compute path lengths with variations to calculate similarity see Rada et al, Hirst & St-Onge, Sussna and Leacock & Chodorow [36, 17, 47, 24]. However, none of these adequately solves the problems inherent in a hierarchy. Approaches that include corpus analysis to refine similarity calculation (see Resnick, Jiang & Conrath, as well as Lin [37, 38, 19, 25]) report results that come closer to human judgement. These measures however are dependent somewhat on how good the information source is.

Richardson and Smeaton combine the lexical database WordNet with Resnick's measure of similarity to give a semantic similarity measure that can be used as an alternative to pattern matching [39]. They use *synsets*, collocations (connected words) and a hierarchical concept graph (HCG) with semantic pointers to hyponyms/hypernyms and meronyms/holonyms. Edges between concepts are given *weights* and the weight of a link is affected by the density of the HCG at that point, the depth in the HCG and the strength of connotation between the nodes.

Spanoudakis and Constantopoulos [45, 46] have investigated in depth the use of metrics to measure the distance between semantic descriptions of artefacts, particularly those developed at various stages of software development. Their model operates on semantic descriptions of objects using the modelling abstractions - *classification*, *generalization* and *attributes*. Objects are compared by four partial distance functions, which compare objects at different levels of detail. The results of the partial distance functions are aggregated into an *Overall Distance* measure which is then transformed into a *Similarity* measure. This model also introduces a *Salience* function, where salience is defined as the belief that an attribute is dominant, based on a compound of the properties *charactericity*, *abstractness* and *determinance*. It is unclear, however, at what point the salience function is calculated and applied to the similarity measure.

Weinstein and Birmingham measure syntactic correspondence between definitions of pairs of terms. Their work deals with artificial ontologies and not real world complexities as *in the context of real-world applications, it is not possible to calculate the meaning of a term* [50]. Contexts restrict accessibility within an ontological structure and are used to hide concepts and relations (ie a relationship with another concept). Contexts are partially ordered and accessibility among contexts is transitive and non-symmetric. In our model, by having a separate graph for each context, real world complexities can be accommodated, because the context itself, defined by the user, gives meaning to the terms used.

Miller and Yang apply clustering techniques and a discrete distance function to measure distances over interval data where the interval distance measures the degree of association [29]. However, all examples shown are quantitative intervals. Their method assesses whether a *semantically meaningful distance metric is available* in order to *consider those attributes together and apply clustering to the set of attributes*. It is unclear how non-numeric intervals are treated.

Rodríguez and Egenhofer [41] present an approach for semantic similarity across different ontologies based on the matching process of each of the specification components in the entity class representations. The similarity function determines lexical similarities with feature sets (functions, parts, attributes). The similarity function equals the weighted sum of each specification component. The work focuses on entity classes and on comparing distinguishing features in terms of strict string matching between synonym sets that refer to those features. It is interesting to note that when undertaking human testing, the subjects' answers varied on the number of ranks used to classify entity classes.

Rodríguez, Egenhofer and Rugg [42] combine feature mapping with semantic distance calculation to assess semantic similarities. Their model for measuring semantic similarity has a strong linguistic basis and takes into account synonyms and different senses in the use of terms. It also considers component-object relations with properties of asymmetry in evaluation of similarity. Their work outlines a model that assesses similarity by combining feature mapping with a semantic distance measurement defined in terms of the relevance of different features in terms of the distance in a semantic network. The global similarity function is a weighted sum of the similarity values for parts, functions and attributes and yields values between 0 and 1. Context, although recognized as a relevant issue for semantic similarity, is not addressed in this work.

Roddick et al. [40] present a unifying semantic distance model in which a graph-based approach is used to quantify the distance between two data values. This approach facilitates a notion of distance, both as a simple traversal distance and as weighted arcs. Transition costs, as an additional expense of passing through a node, are also accommodated. This model recognizes context as the most important factor in measuring distance.

### 3 Framework

#### 3.1 Overview

The multi-level framework for the analysis of sequential data is comprised of four levels, see Figure 1. Levels One and Two take user defined parameters for all support levels and are not context dependent, since we are dealing with the frequent episodes at a individual character level. On the other hand, Levels Three and Four which are concerned with complete words and phrases are necessarily context dependent and as such allow the interaction by means of an accept and reject policy as well as the ability to alter the current support levels.

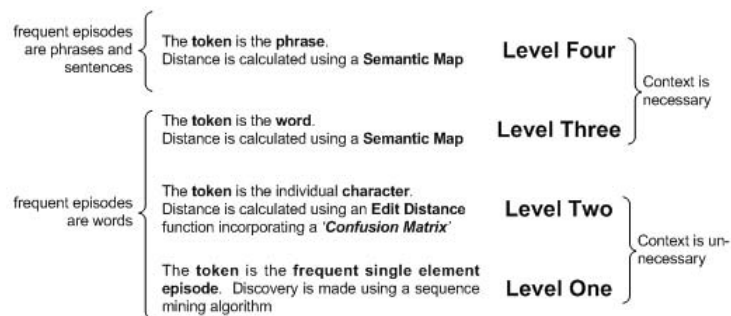


Fig. 1: Overview of the framework

### 3.2 Level One – Discovering the single element episodes

The algorithm we use as the basis for generating the frequent episodes is one which was developed by the authors and was detailed in a previous paper [30], however some modifications have been made to accommodate for the fact that we are now primarily interested in using text documents as input. Having said this the following definitions are still applicable.

Let the set of available input tokens (the alphabet), denoted  $T$ , be defined as  $T = \langle t_1, \dots, t_k \rangle \mid t_i \neq t_j, i \neq j, 1 \leq i, j \leq k$ . A sequence  $S$  is then defined as a time ordered ( $<$ ) sequence of input tokens and is denoted  $S = \langle s_1, s_2, \dots, s_m \rangle \mid s_i \in T, 1 \leq i \leq m$ . An *episode*, denoted  $E$ , is a sequence of tokens,  $\langle s_n, s_{n+1}, \dots, s_{n+k} \rangle$ , where  $E \subseteq S$ .

The user defined *lookahead*,  $l$  (similar to Mannila *et al.*'s window concept[27]), defines the maximum length episode to be mined, where  $|E| \leq l \leq |S|$ . A window, denoted  $w$ , is defined as the length of  $E$ , where  $|E| \leq l$ , at any point during the mining process. Therefore the maximum number of windows, *max\_win* is given by  $|S| - w + 1$  and the *frequency* of  $E$  in  $S$  is defined as the number of windows in which  $E$  appears. The minimum frequency required for an episode to be reported, *min\_freq* denoted  $\delta$ , is calculated using a support,  $\sigma_1$  (user defined), multiplied by *max\_win* at any given point in the mining run.

In addition to this *lookahead* the provision for a *delimiter\_list* has been introduced; the period (.) and the comma (,) being two such examples, and these override the *lookahead* at any time they are encountered, even if it is before the maximum value of the *lookahead* has been reached. However, if a delimiter is reached, then the length of  $E$  when the delimiter was reached becomes the the window,  $w$ , for the purpose of calculating *min\_freq*,  $\delta$ .

Thus our problem for this level is to find all single element candidate episodes

$$E_i \text{ on } \{S \mid |E_i| \leq l \text{ or } [delimiter\_list], freq(E_i) \geq \delta, \delta = (|S| - w + 1) \times \sigma\}$$

and have them available for Level Two.

### 3.3 Level Two – Edit distance

There are two possible strategies for dealing with the incorporation of the edit distance metric for the removal of typographic errors; 1) as a pre-process data cleaning step or, 2) after the single element candidate episodes have been generated. There may be some argument with respect to which strategy is best<sup>3</sup>, but in the context of this work, and since the methodology is the same regardless of the positioning, we have adopted the latter. This results in the need for two processes to be implemented on the generated single element candidate episodes:

- 1.) Edit distance calculation and merge
- 2.) Semantic distance calculation and merge

<sup>3</sup> time, usefulness of the cleaned data for other purposes etc.

In this Level we are concerned with the edit distance calculation and it is performed as follows. The current value of support,  $\sigma_1$ , is used to collect those episodes that meet threshold, and another list is collected for those episodes that fall between the primary support value and secondary support value,  $\sigma_2$ . A third support value,  $\sigma_3$ , is used for a cut-off that indicates those candidates that are not considered in any of the calculations<sup>4</sup>. Once this list has been compiled the edit distance algorithm, incorporating the *confusion matrix*, is used as per Algorithm 3.1. In traditional mining tasks it is normal to have one support threshold,  $min\_supp$  or  $\theta$ , or in the case of hierarchical association mining it may be that there are two or more on a sliding scale. We have introduced a sliding support scale based on the lengths of the episodes being mined and this has proved to be useful in detecting more interesting longer episodes. However, when dealing with context dependent measures it has become apparent that it may be necessary to have three *persistent* levels of support<sup>5</sup>. These levels we have named:

- support,  $min\_supp$  ( $\theta$ ), which is the same as any traditional support heuristic,
- distance support,  $dist\_supp$  ( $\theta_\alpha$ ). Currently this is calculated to be one standard deviation from  $min\_supp$ . The purpose of this heuristic is that those elements that are between  $min\_supp$  and  $dist\_supp$  will be used as the seeds for any distance calculations that are performed, and
- lower bound support,  $low\_supp$  ( $\theta_\beta$ ). Currently this is calculated to be two standard deviations from  $min\_supp$ . Any elements that fall below this threshold will not be included in any distance heuristic and furthermore only those elements that fall between  $dist\_supp$  and  $low\_supp$  will be used in either Level Two, Three or Four distance calculations.

The calculations of the standard deviation can be made after the first pass of the data when the total number of elements, the number of unique tokens<sup>6</sup> and the frequency of the elements is known. The standard deviation is calculated as:

$$\sigma = \sqrt{\frac{\sum x^2}{n} - \bar{x}^2}, \text{ and therefore, } \theta_\alpha = \theta - \sigma \text{ and } \theta_\beta = \theta - 2\sigma$$

Some limitations of using supports based on standard deviations are:

- 1.) If the  $min\_supp$  has to be set very low, as is the case with our current synthetic datasets, then the values are meaningless. (Our current synthetic datasets use a  $min\_supp$  that is just above  $3\sigma$ )
- 2.) the lower bound,  $low\_supp$  may have to be cut at zero dependent on the value of  $min\_supp$  (Negative values make no sense here)

<sup>4</sup> In this work we have used a static value for  $\sigma_2$  that is 5% lower than that of  $\sigma_1$  and a value that is 15% lower than  $\sigma_1$  for  $\sigma_3$ . However, for datasets that are more normally distributed values that are multiples of the standard deviation of the dataset can be used, i.e. 1SD for  $\sigma_2$  and 2SD for  $\sigma_3$ .

<sup>5</sup> The term *persistent* here relates to the fact that they are always there, not that they are always the same value.

<sup>6</sup> Here the term tokens is used to represent the alphabet

---

**Algorithm 3.1** Pseudo code for using string edit distance

---

**Input:** a list  $L$ , of episodes  $e$  and a support  $\sigma_1$ , a support  $\sigma_2$  and a minimum edit distance,  $\epsilon$ .

**Output:** the collection of frequent single element episodes  $E$ .

```

1: for all  $\alpha$ , in  $L$  do
2:   for all  $\sigma_3 < \beta < \sigma_2$  do
3:     calculate the edit distance,  $\rho$ , between  $\alpha$  and  $\beta$ 
4:     if  $\rho \leq \epsilon$  then
5:       merge  $\alpha$  and  $\beta \Rightarrow \alpha$ 
6:       frequency  $\alpha = \text{frequency}(\alpha + \beta)$ 
7:       if the support for the merged  $\alpha$  and  $\beta$  is  $\geq \sigma_1$  then
8:         remove both  $\alpha$  and  $\beta$  from further processing
9:       end if
10:    end if
11:  end for
12: end for

```

---

After this processing we have a set of frequent one element episodes,  $E$ , that has eliminated the majority, if not all, typographic errors. The choice of the value for  $\epsilon$ , see Algorithm 3.1 line 4, can be varied to accommodate a more or a less strict interpretation of what a typographic error is, but during this study we 'hard coded' this value to be relatively small and therefore, two words were considered the same only if they contained a single typographic error. A more exhaustive study using documents from varied domains may show that this value should be a user-defined parameter to the mining process that would take into account the domain knowledge of the users of the system.

This set of frequent one element episodes can now be processed in Level Three using the semantic distance measures.

### 3.4 Level Three – Semantic distance between words

In this work, we use the unifying semantic distance model developed by Roddick et al. [40] for its flexibility and ease of use. A separate semantic map is constructed for each context. The semantic map is stored in a directed graph with words (or phrases) in nodes and arcs connecting them weighted with agreed values. The graph may be populated using a dictionary, thesaurus or ontology with the weights calculated by any of the previously mentioned distance measures, see Section 2.3, or by a domain expert. A value  $d(n_i, n_j)$ , of between 0 and 1, representing the distance between each adjacent node is associated with each directed arc indicating the uni-directional or bi-directional distance between the nodes.

A distance may be calculated for any word or phrase pair, or alternatively, a set of words/phrases can be retrieved, each of which is deemed to be *close to* a given word by specifying a threshold value. Transitivity of semantics is enabled by arc weights, transition costs and a focussing factor. The number of arcs does not automatically increase distance as arc weights may be zero. The 'shortest

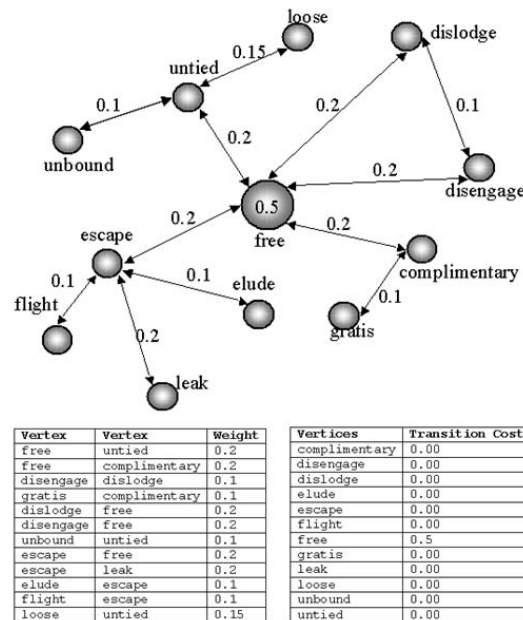


Fig. 2: A portion of the Semantic Map surrounding the term ‘**free**’, with the associated input files below it.

path' algorithm is used to calculate distances between nodes in a fully connected graph.

Words are easily added to the map by specifying an additional node and at least one arc connecting it to an existing node, see Figure 2. This allows abbreviations, slang and jargon terms to be included in subsequent mining, fully utilising the re-usability of the map.

To process the list of frequent one element episodes the same approach to that used in Algorithm 3.1 can be used. The choices for the cut-offs, see Line 2, can remain the same or be changed depending on the requirements of the user. Further to this, since this level is interactive, the user may or may not accept the terms presented for *compression* into a single term, for the purpose of meeting the support threshold.

*Example 3.*

TERM	TERM
<b>bush</b> support 48% <i>close to</i>	<b>president</b> support 32%
with a distance value of 0.05	

- If the distance value is less or equal to  $\epsilon$ , (Line 4 - Algorithm 3.1), then the ‘compression’ is automatic,

- Else if the distance value is greater than  $\epsilon$ , then the user is prompted to ‘compress’ the two terms if they agree that they are synonyms. The second of the two terms, in this instance **president**, is displayed with its surrounding words from the document to assist with this decision.
- Else the term rejected and the next term is assessed

The above example, Example 3, illustrates the degree of control on the finer points of semantic relevance that the user can impose, thus enhancing the usability of our framework.

### 3.5 Level Four – Semantic distance between phrases

At this time we have not implemented this level, but we envisage that the processing would be very similar, if not the same, as Level Three. Any differences would be as a result of the type of Semantic Map that would be employed. The difficulty for this level is not necessarily in the processing phase, but rather in the creation and maintenance of any ‘phrasal’ semantic maps that are used. Currently we are investigating the use of ontologies for this purpose.

## 4 Discussion and Future Work

In this paper we have outlined a framework to mine text as a sequence of tokens in the first instance and to incorporate the detection and processing of, in the second instance typographic errors and in the third instance semantically related terms.

There are however some enhancements that can be made especially using the semantic distance of terms. At present we are only applying the semantic distance to single element episodes but there is no reason why the maps can not be expanded to handle phrases that are semantically similar and then these phrases can be processed as single element episodes. One benefit of such an approach is that the mining process becomes more tractable since potentially we would not be generating long sequences<sup>7</sup>.

In addition to the above we are currently implementing the algorithms to fully realise the framework so that exhaustive tests can be conducted to validate and verify the concept.

## References

1. A.V. Aho. Algorithms for finding patterns in strings. In J van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume A: Algorithms and Complexity. Elsevier, 1990.

---

<sup>7</sup> This benefit would be more fully realised when using a candidate generation and prune approach.

2. Helena Ahonen, Oskari Heinonen, Mika Klemettinen, and A. Inkeri Verkamo. Applying data mining techniques in text analysis. Tech Report C-1997-23, University of Helsinki, Department of Computer Science, 1997.
3. Helena Ahonen, Oskari Heinonen, Mika Klemettinen, and A. Inkeri Verkamo. Applying data mining techniques for descriptive phrase extraction in digital document collections. In *Proceedings of the Advances in Digital Libraries Conference*, page 2. IEEE Computer Society, 1998.
4. Amihoud Amir, Moshe Lewenstein, and Ely Porat. Faster algorithms for string matching with  $k$  mismatches. In *Proceedings of the Eleventh annual ACM-SIAM Symposium on Discrete Algorithms*, pages 794–803, San Francisco, California, United States, 2000. Society for Industrial and Applied Mathematics.
5. Abdullah N. Arslan and Ömer Egecioglu. An efficient uniform-cost normalized edit distance algorithm. In *6th Symposium on String Processing and Information Retrieval (SPIRE'99)*, pages 8–15. IEEE Comp. Soc, 1999.
6. Abdullah N. Arslan and Ömer Egecioglu. Efficient algorithms for normalized edit distance. *Journal of Discrete Algorithms*, 1(1):3–20, 2000.
7. Tuğkan Batu, Funda Ergün, Joe Kilian, Avner Magen, Sofya Raskhodnikova, Ronitt Rubinfeld, and Rahul Sami. A sublinear algorithm for weakly approximating edit distance. In *Proceedings of the Thirty-Fifth ACM Symposium on Theory of Computing*, pages 316–324, San Diego, CA, USA, 2003. ACM Press.
8. Jon L. Bentley and Robert Sedgewick. Fast algorithms for sorting and searching strings. In *Proceedings of the Eighth annual ACM-SIAM Symposium on Discrete Algorithms*, pages 360–369, New Orleans, Louisiana, United States, 1997. Society for Industrial and Applied Mathematics.
9. Rajman and Besançon. Text mining — knowledge extraction from unstructured textual data. In *6th Conference of International Federation of Classification Societies (IFCS-98)*, Rome, 1998.
10. D. Breslauer and L. Gąsieniec. Efficient string matching on coded texts. In Z. Galil and E. Ukkonen, editors, *Proceedings of the 6th Annual Symposium on Combinatorial Pattern Matching*, pages 27–40, Espoo, Finland, 1995. Springer-Verlag, Berlin.
11. H. Bunke and J. Csirik. Edit distance of run-length coded strings. In *Proceedings of the 1992 ACM/SIGAPP Symposium on Applied Computing*, pages 137–143, Kansas City, Missouri, United States, 1992. ACM Press.
12. Sarah Chan, Ben Kao, C. L. Yip, and Michael Tang. Mining emerging substrings. Tech Report TR-2002-11, HKU CSIS, 2002.
13. Richard Cole and Ramesh Hariharan. Approximate string matching: a simpler faster algorithm. In *Proceedings of the Ninth annual ACM-SIAM Symposium on Discrete Algorithms*, pages 463–472, San Francisco, California, United States, 1998. Society for Industrial and Applied Mathematics.
14. Graham Cormode and S. Muthukrishnan. The string edit distance matching problem with moves. In *Proceedings of the Thirteenth annual ACM-SIAM Symposium on Discrete Algorithms*, pages 667–676, San Francisco, California, 2002. Society for Industrial and Applied Mathematics.
15. Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.
16. Patrick A. V. Hall and Geoff R. Dowling. Approximate string matching. *ACM Computing Surveys*, 12(4):381–402, 1980.
17. G. Hirst and D. St-Onge. Lexical chains as representations of context for the detection and correction of malapropisms. In C. Fellbaum, editor, *WordNet: An Electronic Lexical Database*, pages 305–332. MIT Press, Cambridge, MA, USA, 1998.

18. Heikki Hyvärö. A bit-vector algorithm for computing levenshtein and damerou edit distances. *Nordic Journal of Computing*, 10(1):29–39, 2003.
19. J. J. Jiang and D. W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *International Conference on Research in Computational Linguistics*, pages 19–33, Taiwan, 1997.
20. Z. Kedad and E. Métais. Dealing with semantic heterogeneity during data integration. In J. Akoka, B. Mokrane, I. Comyn-Wattiau, and E. Métais, editors, *Eighteenth International Conference on Conceptual Modelling*, volume 1728 of *Lecture Notes in Computer Science*, pages 325–339, Paris France, 1999. Springer.
21. H. Kozima. Text segmentation based on similarity between words. In *31st Annual Meeting of the Association for Computational Linguistics*, pages 286–288, 1993.
22. H. Kozima and T. Furugori. Similarity between words computed by spreading activation on an english dictionary. In *6th Conference of the European Chapter of the Association for Computational Linguistics*, pages 232–239, Utrecht, Netherlands, 1993.
23. Gad M. Landau, Eugene W. Myers, and Jeanette P. Schmidt. Incremental string comparison. *SIAM Journal on Computing*, 27(2):557–582, 1998.
24. C. Leacock and M. Chodorow. Combining local context and wordnet similarity for word sense identification. In C. Fellbaum, editor, *WordNet: An Electronic Lexical Database*, pages 265–283. MIT Press, Cambridge, MA, USA, 1998.
25. D. Lin. An information-theoretic definition of similarity. *Proc. 15th International Conf. on Machine Learning, Morgan Kaufmann, San Francisco, CA*, pages 296–304, 1998.
26. H. Mannila and H. Toivonen. Discovering generalized episodes using minimal occurrences. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96)*, pages 146–151, Portland, Oregon, 1996. AAAI Press.
27. Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1(3):259–289, 1997.
28. G. A. Miller and W. G. Chalres. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28, 1991.
29. R.J. Miller and Y. Yang. Association rules over interval data. In Joan Peckham, editor, *ACM SIGMOD Conference on the Management of Data*, pages 452–461, Tucson, Arizona, USA, 1997. ACM Press.
30. Carl H. Mooney and John F. Roddick. Mining relationships between interacting episodes. In Michael W. Berry, Umeshwar Dayal, Chandrika Kamath, and David Skillicorn, editors, *Proceedings of the Fourth SIAM International Conference on Data Mining*, Lake Buena Vista, Florida, 2004. SIAM.
31. J. Morris and G. Hirst. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–48, 1991.
32. Gonzalo Navarro. A guided tour to approximate string matching. *ACM Computing Surveys*, 33(1):31–88, 2001.
33. M. Okumura and T. Honda. Word sense disambiguation and text segmentation based on lexical cohesion. In *15th Conference on Computational Linguistics*, volume 2, pages 755–761, Kyoto, Japan, 1994.
34. B. J. Oommen and R. K. S. Loke. Pattern recognition of strings with substitutions, insertions, deletions and generalized transpositions. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, volume 2, pages 1154–1159, 1995.

35. B. J. Oommen and K. Zhang. The normalized string editing problem revisited. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(6):669–672, 1996.
36. R. Rada and H. Bicknell. Ranking documents with a thesaurus. *Journal of the American Society for Information Science (JASIS)*, 40(5):304–310, 1989.
37. P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *14th International Joint Conference on Artificial Intelligence*, , pages 448–453, Montreal, 1995.
38. P. Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11:95–130, 1999.
39. R. Richardson, A.F. Smeaton, and J. Murphy. Using wordnet as a knowledge base for measuring semantic similarity between words. Technical Report Working Paper CA-1294, School of Computer Applications, Dublin City University, 1994.
40. J. F. Roddick, K. Hornsby, and D. De Vries. A unifying semantic distance model for determining the similarity of attribute values. In M. Oudshoorn, editor, *26th Australasian Computer Science Conference (ACSC2003)*, volume 16, pages 111–118, Adelaide, Australia, 2003. ACS.
41. M. A. Rodríguez and M. J. Egenhofer. Putting similarity assessment into context: Matching-distance with the user’s intended operations. In P. Bouquet, L. Serafini, P. Brézillon, M. Benerecetti, and F. Castellani, editors, *2nd International and Interdisciplinary Conference on Modeling and Using Context, CONTEXT-99*, volume 1688 of *Lecture Notes in Artificial Intelligence*, pages 310–323, Trento, Italy, 1999. Springer.
42. M.A. Rodríguez, M.J. Egenhofer, and R.D. Rugg. Assessing semantic similarities among geospatial feature class definitions. In A. Vekovski, K. Brassel, and H.-J. Schek, editors, *Second International Conference on Interoperating Geographic Information Systems, INTEROP’99*, volume 1580 of *Lecture Notes in Computer Science*, pages 189–202, Zurich, Switzerland, 1999. Springer.
43. H. Rubenstein and J. B. Goodenough. Contextual correlates of synonymy. *Computational Linguistics*, 8(10):627–633, 1965.
44. David Sankoff and Joseph B. Kruskal. *Time warps, string edits, and macromolecules / The theory and practice of sequence comparison*. David Hume series. Center for the Study of Language and Information, Stanford, Calif., reissue ed. edition, 1999.
45. G. Spanoudakis and P. Constantopoulos. Similarity for analogical software reuse: A computational model. In *11th European Conference on Artificial Intelligence (ECAI ’94)*, pages 18–22, Amsterdam, The Netherlands, 1994.
46. G. Spanoudakis and P. Constantopoulos. Elaborating analogies from conceptual models. *International Journal of Intelligent Systems*, 11(11):917–974, 1996.
47. M. Sussna. Word sense disambiguation for free-text indexing using a massive semantic network. In *Second International Conference on Information and Knowledge Management*, pages 67–74, Arlington, Va, USA, 1993.
48. Walter F. Tichy. The string-to-string correction problem with block moves. *ACM Transactions on Computer Systems (TOCS)*, 2(4):309–321, 1984.
49. Robert A. Wagner and Michael J. Fischer. The string-to-string correction problem. *Journal of the ACM (JACM)*, 21(1):168–173, 1974.
50. P.C. Weinstein and W.P. Birmingham. Agent communication with differentiated ontologies: eight new measures of description compatibility. Technical report, Department of Electrical Engineering and Computer Science, University of Michigan, 1999.



# Building a Hierarchical Hidden Markov Model

## An application to health insurance data

Ah Chung Tsoi<sup>1</sup>, Shu Zhang<sup>2</sup>, and Markus Hagenbuchner<sup>2</sup>

<sup>1</sup> Australian Research Council, GPO Box 2702 Canberra, ACT 2601

<sup>2</sup> Faculty of Informatics, University of Wollongong, NSW 2522

**Keywords:** Health data mining, clustering and classification, pattern recognition, automatic annotation of data.

**Abstract.** In this paper, we describe a novel data driven hierarchical hidden Markov model. We describe a recursive two step process: using the Gaussian mixture technique to cluster profiles extracted from the raw data into groups, and then apply the hidden Markov model to further group profiles with similar temporal behaviors into sub-clusters. The two step process is then recursively applied until a hierarchical hidden Markov model is constructed. This methodology is applied to health insurance data. It is found that profiles with similar temporal behaviors are grouped into the same cluster. It is also found that by using a priori information on the profiles, we can further distinguish ambiguous profiles into ones which may be suffering from particular diseases. Thus, the methodology can be used for large scale automatic annotation of databases.

## 1 Introduction

With the rapid development in computers, networks and database technologies, data collection and storage is becoming an almost effortless task. However, the interpretation and extraction of knowledge from the data collected is still in its infancy. There are a number of ways in which knowledge can be extracted from the data. For example, one may extract some underlying rules in which data is associated, or one may cluster data into groups of similar objects.

In this paper, we are concerned with the grouping together of temporal sequences which have similar behaviors. A temporal sequence is a collection of an ordered sequence of events with embedded temporal dependence. The aim of this paper is to investigate ways in which similar temporal behavioral patterns hidden in a very large set of medical transaction data can be grouped together. It will be shown that with judicious use of limited a priori information, similar temporal behavioral patterns which correspond to patients suffering from particular ailments can be distinguished. This is interesting in that the grouping of temporal patterns together without any a priori information forms groups which are unlabelled. But by providing very limited a priori information, in this case, the treatment information of the medical problems suffered by patients <sup>3</sup>, it is possible to further disambiguate similar temporal behavioral patterns

---

<sup>3</sup> Note that we do not have access to notes taken by medical service providers, nor do we have access to the diagnosis of the patient's ailments. The information on medical condition is inferred from the types of procedures taken by the medical service providers.

within an unlabelled group concerning the nature of the patient's ailments. Thus, the proposed technique can be used for large scale automatic annotation of databases containing unlabeled data. Such automatic annotation will add value to the existing medical transaction databases, routinely collected by medical insurance vendors. This paper will present the ideas in a procedural fashion, and will make use of data from large medical transaction databases.

A set of 180 GB de-identified medical records, which covers 7 consecutive quarters of transactions, has been provided by the Australia Health Insurance Commission (HIC), a national medical insurance vendor which provides universal health cover to all Australian citizens and permanent residents, with the aim of discovering similar temporal behavioral patterns amongst patients in the dataset. The dataset contains detailed information for each medical claim made by patients, from personal details such as name<sup>4</sup>, gender, age and address<sup>5</sup>, to medical service details such as the name and address of the provider<sup>4</sup>, the date and the type of service<sup>6</sup>, the amount of benefit paid to the service provider along with the payment method (pay at service, "direct billing"), etc.

The ideas presented in this paper can be subdivided into the following five tasks:

- (1) **Feature extraction** to extract representative features of the underlying temporal behavior of the patient.
- (2) **Creation of cohorts** considers the fact that patient's medical records change dramatically with age. Patients of similar age form an age cohort. Steps (1) and (2) are pre-processing steps which prepare the data for the application of pattern discovery techniques.
- (3) **Clustering** Gaussian mixture clustering is applied to identify clusters among the data for each age cohort; the result of this clustering serves the purpose of labeling of data. This step gathers profiles together according to some norm based on the entire profile, rather than based on temporal variations within the profile.
- (4) **Pattern discovery** is accomplished through hidden Markov models (HMM), a stochastic model commonly used in temporal behavioral pattern discovery, is proposed for the task at hand. HMMs are not particularly suited for large scale data mining tasks given the high computational demand for training these models. This problem is alleviated through the recursive training of HMMs on small portions of the data set; a step which will be addressed in greater detail later in this paper. Steps (3) and (4) are executed recursively until convergence occurred. This recursive process yields a set of HMMs which are hierarchically clustered.
- (5) **Automatic annotation** . Without any a priori information, the clusters will contain temporal behaviours of patients who might be suffering from various ailments. In other words, different ailments will produce similar temporal behavioral patterns. In order to disambiguate these similar patterns, limited a priori information on the medical treatments received by patients in the form of items used by the medical

<sup>4</sup> This has been replaced by a unique identifier which bears no resemblance to the name as a means of protecting the identity.

<sup>5</sup> Only the post code is provided to ensure privacy.

<sup>6</sup> HIC produces a Medical Benefit Schedule which encodes each medical service with a unique item number, together with the amount of benefit a patient can claim for that particular service.

service providers is used to further separate the patterns within the same cluster as obtained in steps (3) and (4) into ailments which the patient might be suffering<sup>7</sup>. Thus, by utilizing the information on medical items used by the medical service providers, we are able to obtain an automatic annotation technique which can annotate large scale medical transaction records. Such annotation would be useful for further research on the databases, e.g., for public health purposes.

The organization of this paper is as follows: Section 2 describes the techniques used in the pre-processing steps. A general discussion of pattern discovery is given in Section 3. Our methodology will be presented in Section 4. Experimental results are shown in Section 5. Conclusions are drawn in Section 6.

## 2 Representation of temporal sequences

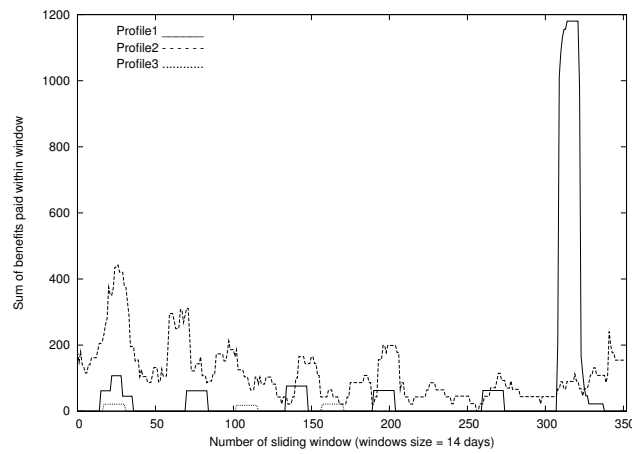
The dataset provided by Health Insurance Commission is a very detailed record of patients' medical history over 7 quarters. The extraction of many of the features contained in the dataset could result in a good representation of temporal sequences. We focus our interest on finding patterns which describe the distribution of the amount of benefit paid, since this feature not only bears patient's medical behaviors, it also encodes most types of service implicitly. Thus, the amount of benefit paid is chosen as the feature to be extracted from the dataset.

When the daily benefit paid is used as the element of temporal sequences, it could bring too much fluctuations into the profile since not many patients see medical service providers daily. A method commonly adopted to smoothing the profile is to use a temporal sliding window. With trial and error, we found 14 days to be a suitable choice for the size of sliding window. This choice could be justified *ex poste* in that a patient is often required to see medical service providers a number of times over a short period of time in a course of treatments. Thus by summing the benefit paid over a time window of 14 days, the natural process of medical treatment can be captured.

A proper representation of the dataset is fundamental for a successful pattern discovery. We improve on the procedure by considering the border effects, which arise as the benefit claims are often not lodged on the same day as the date of service. We observed that a claim could be made as late as several months after the provision of the service. When using the entire 7 quarters of data available to us could introduce artificial border effects, as in the last quarter of data, some of the claims may not have been lodged since some patients may not lodge them until much later<sup>8</sup>. This distortion can be overcome by introducing a cut-off date for claims made within a particular period of time. We chose 365 days or one year as the period for each patient as it was observed that this ensures that 99.8% of services provided are claimed. As a result, a one-dimensional profile of 352 elements is obtained for each patient by extracting, from the longitudinal medical records, the benefits paid over 365 days and then totaling the value by sliding a window of size of 14 days with a sliding step of 1. Three typical profiles are shown in Figure 1.

<sup>7</sup> Note that we do not have access to medical service providers' diagnosis nor notes.

<sup>8</sup> The legal limit is that any claim must be made within 12 months of the service being rendered.



**Fig. 1.** Three profiles from the dataset.

It is found that a patient's medical condition can change dramatically with age. Segmenting the profiles into cohorts of patients of similar age gives two advantages: First, it reduces the size of the dataset for later tasks, and secondly it reduces the impact of age related medical issues on the learning tasks. Consequently, we generated age-cohorts as illustrated in Table 1.

Table 1 shows that for both male and female patients the amount of benefit paid and the frequency of medical services decrease until the patients reach adolescence. Then these values experience a rapid increase before slowing down until retirement age for female patients; a similar trend can be observed for male patients but at a much slower rate. The striking observation is that female patients older than 16 years of age use medical services considerably more frequently than their male counterpart. This behavior may come from the sexual differences and is associated with its ongoing consequences (e.g. child bearing). In contrast, more boys (male children younger than 16) seek medical services than girls of the same age. This may be caused by the general ob-

**Table 1.** Cohort groups and sizes.

Age cohort	Female			Male		
	Number of patients	Av. num. of claims	Av. value of claim	Number of patients	Av. num. of claims	Av. value of claim
0-3	460,548	7.65	229.94	489,403	8.29	257.56
4-6	347,328	5.58	166.34	368,972	5.79	179.33
7-15	957,225	4.47	158.04	997,945	4.39	161.58
16-24	1,053,985	7.41	308.14	946,464	4.77	196.22
25-35	1,498,662	9.04	437.98	1,257,976	5.29	227.97
36-44	1,221,511	8.59	448.24	1,074,617	6.09	285.41
45-55	1,287,334	9.82	524.06	1,199,845	7.38	388.57
56-70	1,101,022	12.73	684.27	1,068,311	11.21	654.19
>70	862,858	16.97	875.29	556,972	15.07	715.07

ervation that boys are more exploratory outdoors and thus incurring a higher exposure to risky activities than girls. This difference fades with aging, when more frequent and expensive medical services are incurred.

In summary, a dimension reduction of the dataset is achieved by setting up a one-dimensional profile for each patient, and the dataset is segmented into age cohorts. The pattern discovery techniques will be applied to each age cohort separately so that patterns in the same age cohort may have a better chance of being grouped together.

### 3 Pattern discovery techniques

This section gives a general introduction to Gaussian Mixture models and Hidden Markov models which will be employed to the pattern discovery task later in this paper. We will also describe a methodology to obtaining a hierarchical set of hidden Markov models from the set of patient profiles. This is a two step process:

**Step 1** Clustering of the entire profiles. In this step, we will group profiles together according to their overall pattern. This is achieved by using a clustering technique such as a Gaussian mixture model.

**Step 2** Use a hidden Markov model to further divide the clusters obtained in Step 1 into clusters of similar temporal behavioral patterns.

#### 3.1 Gaussian mixture model as a clustering tool

The task of discovering temporal patterns could be initialized by clustering because of its ability to segment data into clusters according to a similarity criterion. There are a number of possible algorithms which can be employed for this purpose, e.g., K-means clustering algorithm [4], Gaussian mixture algorithm [5], mixture of HMMs [10], self organizing map method [6]. Due to its simplicity the K-means algorithm has been used in first experiments, but the results showed that it does not cluster sparse vectors well. In our case, sparse vectors are common occurrences as most patients do not visit a medical service provider regularly at daily intervals. Mixture of HMMs can serve as a tool of clustering, but given the size of the dataset, it will be computationally expensive to use. Similarly the same observation is true for the Self-Organizing Map. In this respect, Gaussian mixture (GM) models are more practical [3].

We assume that the data observed are generated by a  $\mathcal{D}$ -dimensional GM of  $N$  components ( $N$  clusters) with the following probability density function [3]:

$$f(\mathbf{x}|\Psi) = \sum_{n=0}^{N-1} p_n g(\mathbf{x}|\Theta_n)$$

where  $\Psi$  denotes the vector encompassing all the mixture parameters ( $\Theta_n$  and  $p_n$ ),  $p_n$  ( $\geq 0, n=0, 1, 2, \dots, N-1, \sum_{n=0}^{N-1} p_n = 1$ ) is the weight of the  $n^{th}$  component in the model, or the possibility that the pattern  $\mathbf{x}$  was generated from the  $n^{th}$  component.  $\Theta_n$  stands for all the parameters ( $\mu_n$  and  $\mathbf{V}_n$ ) of the  $n^{th}$   $\mathcal{D}$ -variate Gaussian distribution  $g(\mathbf{x}|\Theta_n)$  of probability density function:

$$g(\mathbf{x}|\Theta_n) = \frac{\exp^{-\frac{1}{2}(\mathbf{x}-\mu)^T(\mathbf{V})^{-1}(\mathbf{x}-\mu)}}{\sqrt{(2\pi)^D \det(\mathbf{V}_n)}} \quad (1)$$

with mean vector  $\mu_n$  and covariance matrix  $V_n$ .

In general, the parameters of the model are typically trained using expectation maximization (EM) algorithm [3] which is used to produce the maximum likelihood estimates  $\hat{\mu}_n$  and  $\hat{V}_n$  to parameters  $\mu_n$  and  $V_n$ , respectively. More importantly, the parameters can be improved through some EM iterations [7] with respect to the dataset  $\mathbf{x} = \{\mathbf{x}_m, m=0, 1, \dots, M-1\}$  so that the model will fit the data better. The updating of parameters in the  $k$ -th EM iteration is as follows [7]:

$$\begin{aligned} h_n^{(k)}(\mathbf{x}_m) &= \frac{p_n^{(k-1)}(\mathbf{x}_m)g(\mathbf{x}_m, \Theta_n^{(k-1)})}{f^{k-1}(\mathbf{x}_m)} \\ \hat{p}_n^{(k)}(\mathbf{x}) &= \frac{\sum_{m=0}^{M-1} h_n^{(k)}(\mathbf{x}_m)}{M} \\ \hat{\mu}_n^{(k)}(\mathbf{x}) &= \frac{\sum_{m=0}^{M-1} \mathbf{x}_m h_n^{(k)}(\mathbf{x}_m)}{M \hat{p}_n^{(k)}(\mathbf{x})} \\ \hat{V}_n^{(k)}(\mathbf{x}) &= \frac{\sum (\mathbf{x}_m - \hat{\mu}_n^{(k)}(\mathbf{x})) h_n^{(k)}(\mathbf{x}_m) (\mathbf{x}_m - \hat{\mu}_n^{(k)}(\mathbf{x}))^T}{M \hat{p}_n^{(k)}(\mathbf{x})} \end{aligned}$$

where  $\hat{p}_n^{(k)}(\mathbf{x})$ ,  $\hat{\mu}_n^{(k)}(\mathbf{x})$  and  $\hat{V}_n^{(k)}(\mathbf{x})$  are the maximum likelihood (ML) estimators for the unknown parameters  $p_n$ ,  $\mu_n$  and  $V_n$ .

The EM iteration or the updating of parameters, will stop when one of the following criteria is satisfied:

1. not enough improvement,  $\gamma > 0$  is given by user,

$$1 \leq \frac{L(\hat{\Psi}^{(k)})}{L(\hat{\Psi}^{(k-1)})} \leq 1 + \gamma, \quad \text{where}$$

$$L(\hat{\Psi}^{(k)}) = \sum_{m=0}^{M-1} \log \left[ \sum_{n=0}^{N-1} \hat{p}_n^{(k)} g(\mathbf{x}_m | \hat{\Theta}_n^{(k)}) \right]$$

2.  $k = K$ ,  $K$  is the maximum number of EM iterations.

After the updating of parameters through the EM algorithm, a Gaussian mixture model is able to cluster each  $\mathbf{x}_m$  ( $m=0, 1, 2, \dots, M-1$ ) into a cluster  $c$  if

$$\beta_n(\mathbf{x}) = \log[\hat{p}_n g(\mathbf{x} | \hat{\Theta}_n)] \quad (2)$$

$$c = \arg \max_{n=0, 1, 2, \dots, N-1} \beta_n(\mathbf{x}) \quad (3)$$

With the GM model, a problematic issue is the choice of the number of components  $N$ . The number of components could be assigned based on a knowledge of the dataset, or obtained by applying some information criteria (IC) [8], which provides a rational choice of the number of components based on Bayesian arguments. In our work, we chose the number as 6 through preliminary experiments on the dataset, while IC will be considered in future work.

The work of pattern discovery cannot be done properly by only employing the GM model, because:

1. the EM algorithm would not guarantee a set of parameters which is located at the global minimum of the likelihood function, i.e., the initialization of the parameters has an impact on the results of estimators. There is no doubt the impact will eventually be transferred to the clustering of dataset implicitly.
2. the number of clusters cannot be decided dynamically with respect to the size of dataset, even with the application of an information criterion since an IC only offers a choice based on convergence in probability.

These issues can be overcome, to some extent, through a recursive application of HMMs.

### 3.2 The Hidden Markov Model

A hidden Markov model (HMM) is a system of states with a probabilistic state transition model which can model a given sequence of events.

A number of variants of HMMs are available [9], such as discrete HMM, continuous observation HMM, input-output HMM. In our case, a continuous HMM is deployed.

It is assumed that the observations  $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_T$  are generated by a multivariate probability density function. For simplicity, we will assume that this is generated by a Gaussian mixture as follows:

$$f_{\mathbf{y}|\mathbf{x}}(\xi|i) = \sum_{m=1}^M c_{im} \mathcal{N}(\xi; \mu_{im}, C_{im}) \quad (4)$$

where  $\mathcal{N}(\xi; \mu_{im}, C_{im})$  denotes a Gaussian probability density function with mean  $\mu_{im}$  and covariance matrix  $C_{im}$ . The notation  $f_{\mathbf{y}|\mathbf{x}}(\cdot)$  denotes the probability of observing  $\mathbf{y}$  given the hidden state sequence  $\mathbf{x}$ . The constants  $c_{im}$  are known as mixing coefficients. In order to be a probability density function, we must have  $\sum_{m=1}^M c_{im} = 1$  for  $1 \leq i \leq S$ , and  $S$  is the size of the alphabet (the dimension of the state space).

It is further assumed that the observation probability density functions are generated by a hidden state  $\mathbf{x}$ ,  $\mathbf{x}$  is a  $S$  dimensional vector. This state follows the evolution equation:

$$\mathbf{x}(t+1) = A\mathbf{x}(t) \quad (5)$$

where  $A$  is the state transition matrix, with initial condition  $\mathbf{x}(0) = \pi_0$ . The parameters in the model are then  $\mathcal{M} = \{S, \pi_0, A, \{f_{\mathbf{y}|\mathbf{x}}(\xi|i), 1 \leq i \leq S\}\}$ .

The problem in HMM estimation can be divided into two sub-problems [9]:

1. Given a series of training observations for a given entity, say, a label, how do we train an HMM to represent this label? This problem becomes the finding of a procedure for estimating an appropriate state transition matrix  $A$ , and observation probability density function  $f_{\mathbf{y}|\mathbf{x}}$  for each state.
2. Given a trained HMM, how do we find the likelihood that it produced the incoming observation sequence.

The HMM estimation algorithm is readily available in e.g., [1, 2, 9]. We will use the training algorithm presented in [9]. And for the problem of finding class labels given a set of observations, the Viterbi algorithm [9] is used.

## 4 The methodology

Our approach to the task of pattern discovery is presented in this section. A two-step approach is used which is then applied recursively to obtain a hierarchical set of HMMs. In the first step, GM is engaged to detect clusters among the profiles. The second step trains a set of HMMs, one for each cluster of profiles. To reduce the computational burden, the HMMs are trained recursively on relatively small subsets. In each iteration, the HMMs are trained on a different subset of randomly selected profiles from the cluster until the classification error reaches a minimum.

### 4.1 Gaussian mixture clustering

It was mentioned in Section 2 that cohorts of patients of similar age are considered for the pattern recognition task. In this section we choose age cohort 56-70 to illustrate the approach. There are a total of 2,169,333 profiles in this age cohort; we will divide them into a training data set and a validation data set. The GM clustering approach addressed in Section 3.1 is employed on 91,219 randomly selected profiles from the data pool. We employ this relatively small subset of data rather than the full data set available in this cohort group as it suffices for illustration purposes and it allows us to reduce the turn around time for experiments. Since both, GM and HMM scale linearly with the number of training data, the training time required for the full dataset are easily estimated through a linear adjustment of the training times stated in this paper. It is important to note that the size of this subset has been chosen so that it contains a good representation of the feature domain available in the full data set.

Out of 91,219 profiles, 64,553 will be used for the training process; all remaining profiles are used for the validation purpose. We assume that the dataset is generated by a Gaussian mixture model of  $N = 6$  components. The parameters of the model will be first estimated by utilizing the EM algorithm, and then updated through a few EM iterations as shown in Section 3.1 for a better representation to the dataset. The updating ends when improvement of the log likelihood of the incomplete data is less than 1% or the number of EM iteration reaches to 50. We applied the GM algorithm to segment the 64,553 profiles into 6 clusters as shown in Table 2.

**Table 2.** The clustering results of Gaussian mixture model with 6 components.

Name of Cluster	Number of Profiles
Cluster A	10780
Cluster B	10225
Cluster C	17762
Cluster D	11105
Cluster E	10315
Cluster F	4366
Total	64553

**Table 3.** Classification results by the hidden Markov model.

Class	A	B	C	D	E	F	Total
A	<b>9485</b>	1285	10	0	0	0	10780
B	4055	<b>3516</b>	2224	430	0	0	10225
C	957	7002	<b>9146</b>	657	0	0	17762
D	0	0	2019	<b>5701</b>	2787	598	11105
E	0	0	186	6207	<b>3282</b>	640	10315
F	0	0	0	0	879	<b>3487</b>	4366
$\Sigma$	14497	11803	13585	12995	6948	4725	64553

As indicated in Section 3.1, these clustering results are not reliable for the task of temporal pattern discovery in the sense that it only provides a clustering based on an assumption of the number of components in the Gaussian mixture model. Nevertheless, the approach provides a meaningful segmentation of the profiles as a whole.

#### 4.2 Recursive HMM modeling

As a widely used pattern discovery technique, HMMs are capable of classifying profiles with similar temporal patterns into the same class. The clustering results of GM could serve as training sets so that 6 HMMs, one for each cluster, are trained. However, the computational efforts of training hidden Markov models render them not particularly suitable for data mining tasks, e.g., it takes about 250 minutes to train a HMM on 10,780 profiles using a workstation with a 2GHz XEON processor, and 2 GB RAM. We propose to take a recursive training-recognizing cycle on subsets of data to ease this computational burden:

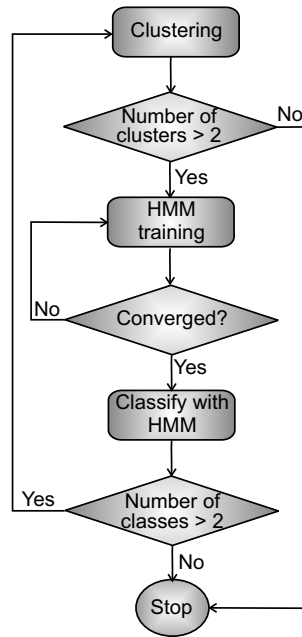
**Training phase:** No more than 2000 profiles are randomly chosen from each cluster to train a corresponding HMM. By doing so it takes about 13 minutes to train a HMM.

**Recognition phase:** All 64,553 profiles are classified by the trained HMMs. The classification re-distributes the profiles between the 6 clusters as shown in Table 3. The rows in Table 3 refer to the clusters obtained by GM whereas the columns are the classifications produced by the set of HMMs. The result will be used for further processing.

The off diagonal numbers of the confusion table shown in Table 3 describe how GM segments different profiles from HMM. For example, the number of profiles in Class E fell from 10,315 to 6,948. Thus, the recognizing phase provides an adjustment to the grouping of profiles by considering temporal patterns discovered in the training set.

The result may not be optimal since only a relatively small number of profiles were engaged during the training process. The quality of HMMs is improved through a recursive application of training-recognition cycle by adopting the grouping results from the recognizing phase of the previous cycle. This iterative process ensures that more and more data from the training set is eventually considered in the training process. The approach is illustrated in Figure 2.

It is observed that the recursion minimizes mis-classification, i.e., the sum of the off diagonal numbers is minimized as shown in Figure 3. The recursive application of training-recognition cycle stops when there are no mis-classifications or when a maximum number of iterations is reached. Here we allow it to run for up to 50 cycles.



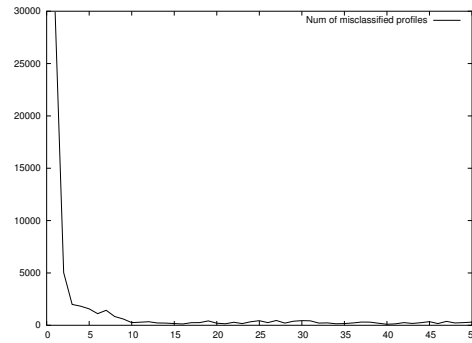
**Fig. 2.** A prototype of our approach of one GM-HMM pattern discovery iteration.

In summary, the recursive training of HMMs significantly reduces the computational burden since it provides a mechanism to detect redundancy in the data set. This is due to the fact that mis-classifications will approach zero faster if the dataset contains many redundant data. The convergence of misclassified profiles indicates that this recursive training of HMMs is a successful way to discover the patterns. The best performance was observed at the 40th iteration when the misclassification reaches a minimum. The resulting confusion matrix is shown in Table 4. It is shown, that the classification of profiles is vastly improved.

### 4.3 Building a hierarchical set of HMMs

In the previous 2 steps of GM-HMM pattern discovery process, we have successfully trained HMMs as representation of patterns. Here we show how the quality of representative of the HMMs could be further enhanced by our refinement process. At the 40-th cycle, even though the 6 HMMs have stretched out their ability to represent the profiles, it is observed that this is still a gross segmentation of the set of profiles since it is impossible to be certain about the number of clusters. The refinement of classification starts with another GM-HMM pattern discovery process for each of the classes obtained so far unless the size of a class is too small to proceed, say, less than 300 profiles<sup>9</sup>. Figure

<sup>9</sup> We set 300 as the threshold since we find that a HMM cannot be properly trained on less than 300 training data. No HMM will be trained on clusters that contain less than 300 profiles. Affected profiles are considered in the recognizing phase to ensure that no profile is discarded.



**Fig. 3.** The convergence of the mis-classification of profiles.

4 explains the process in a visual manner. The first iteration of the GM-HMM process at the top level ends up with 6 classes after 50 iterations of training of HMMs which reaches the best performance at the 40-th cycle. At the 40-th cycle, another iteration of GM-HMM process at sub-level 1 has proceeded to each of the classes (A to F) for further refinement of the classification results. It is noted that instead of 50, there are only 40 iterations of HMMs training conducted in Class A, i.e., the HMMs trained at the 39-th and the 40-th cycles respectively classify the profiles of Class A in the same manner therefore the HMM training stops. The early exit of HMM training indicates that the HMMs trained on the 2000 randomly chosen profiles from the classes are robust, especially when the size of the data set is relatively large. A similar observation can be made on the GM-HMM process to Class F, where only 14 iterations have been run before the misclassification to the profiles converges to zero. Another observation is that the GM algorithm only provides 4 classes to the profiles in Class F rather than the default number of 6. This implies that the GM algorithm is able to reduce the number of clusters, given the default value. As it is pointed out in Section 4.2, the quality of HMMs can be improved through the recursively deployment of training and recognizing cycles to the profiles until the stopping criterion is reached. Then, yet another GM-HMM process is about to start to further refine the classifications when the misclassification reaches a

**Table 4.** Classification results by trained HMMs at the 40th iteration where the number of misclassification reaches a minimum of 110.

Class	A	B	C	D	E	F	Total
A	<b>9030</b>	0	0	0	0	0	9030
B	14	<b>14227</b>	11	0	0	0	14252
C	0	1	<b>16457</b>	0	0	0	16458
D	0	0	64	<b>13952</b>	0	0	14016
E	0	0	0	8	<b>8853</b>	0	8861
F	0	0	0	0	3	<b>1933</b>	1936
$\Sigma$	9044	14228	16532	13952	8856	1933	64553

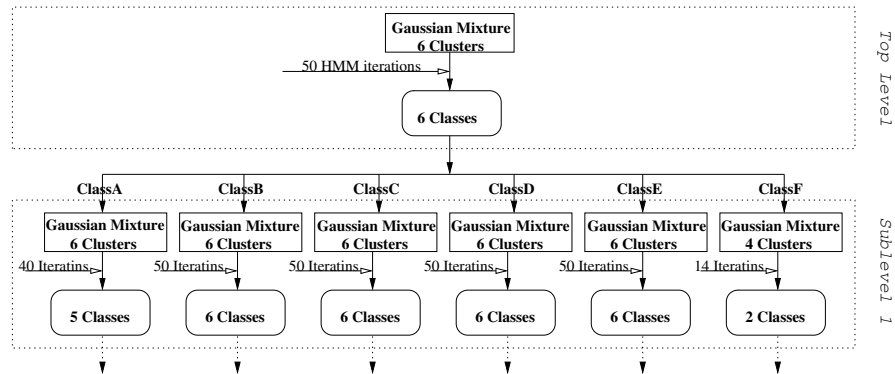


Fig. 4. A flowchart of the iterative GM-HMM refinement process.

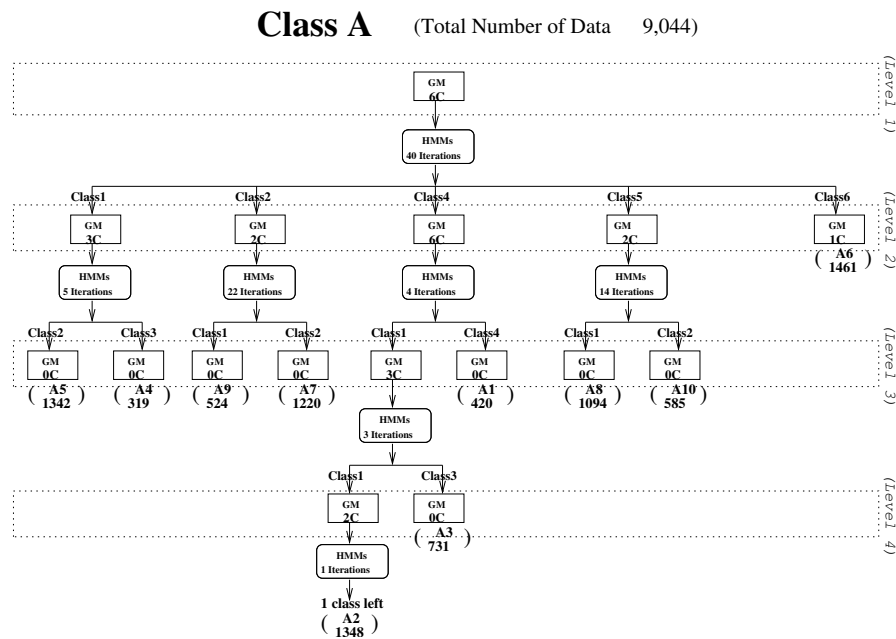
minimum as long as the size of class is large enough. Figure 5 uses Class A to provides a detailed example of the tree-like recursive application of the GM-HMM process

Through this refinement process, we decompose the bigger classes into smaller ones so that each HMM is more specialized on the patterns it stands for. Our tree-like approach of pattern discovery ends up with 76 classes ( $A_1$  to  $A_{10}$ ,  $B_1$  to  $B_{17}$ ,  $C_1$  to  $C_{20}$ ,  $D_1$  to  $D_{16}$ ,  $E_1$  to  $E_{10}$  and  $F_1$  to  $F_3$ ). The average benefit paid for each patient in a class gives the order of subclass names. The higher the average value, the higher the class order. e.g., the average benefit paid for patients in class  $B_1$  is lower than that of class  $B_2$ . But it does not guarantee that the value of a subclass from class A, say  $A_{10}$ , is lower than that of a subclass of B, say, subclass  $B_1$ . The alphabetical order of the classes is decided by the maximum average value of all its subclasses.

## 5 Experiment results

Note that the profiles are unlabeled, and the HMMs were trained in an unsupervised fashion. In order to give a meaning to the classes and patterns represented by HMMs, and in order to assess the quality of the results we extract properties on the patients in the training set and determine how these patients are classified. For example, it is interesting to see how patients suffering different illnesses are distributed over the classes according to the patterns discovered. It would not be feasible to expect that patients suffering the same disease to be classified into the same class since the patterns are clustered based on their temporal medical behaviors rather than on diseases. Throughout the development of a disease, patients suffering the same disease could show different medical behaviors at different stages therefore they do not necessarily share the same pattern or in the same class. On the other hand, patients suffering different diseases could be classified into the same class as long as they share similar temporal behaviors.

The following four experiments demonstrate how patients suffering from different diseases are classified when applying the hierarchical HMM models as indicated in previous sections to the data pool of 91,219 patients. Figure 6 shows the percentage of patients against the classes. They can serve as annotation of the dataset.

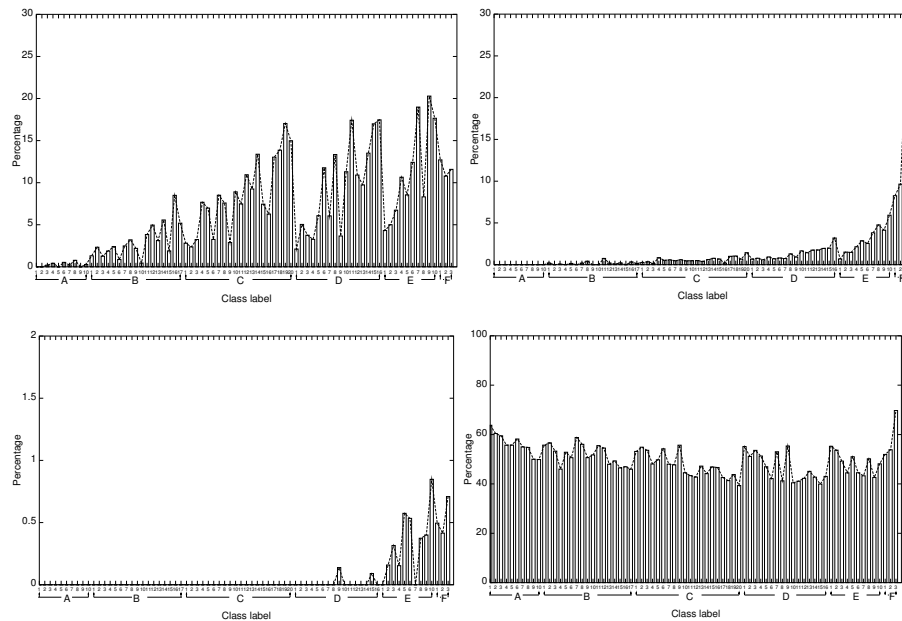


**Fig. 5.** Applying GM-HMM iteratively to profiles in Class A. The values in the brackets give the class label, and the number of profiles in the sub-class..

### 5.1 Experiment 1: Diabetes

There are a few treatments which are defined as diabetes management and exclusively used by diabetics<sup>10</sup>. It is found that 6,138 out of 91,219 patients have gone through such treatments. The classification of these patients are shown on the top-left of Figure 6. Shown is the number of cases (in percents) in each class normalized with respect to the size of the classes so that the effect of obtaining large percentage values for large classes can be avoided. This observation is not surprising since many diseases can be associated with diabetes and hence a patient's medical behaviors can differ significantly. Consequently, we find diabetics classified over a broad spectrum of classes. Seriously ill diabetics who may have experienced different kinds of complications which result in frequent medical visits or some treatments which incur large benefit claims. Consequently they are classified into higher classes (such as subclasses of E and F), while diabetics with situation controlled at a stable stage are classified into the middle classes (like subclasses of C or D) where some other medical treatments which may not be related to diabetes could bring the differences between their behavior patterns. The noticeably low density of diabetic in the lower classes (like the subclasses of A or B) demonstrates that there are courses of medical treatments which have little to do with the medical behaviour of a diabetic. In fact there are 12 classes into which no dia-

<sup>10</sup> Items 66551, 66557, 66319, 66322 as defined in Medicare Benefits Schedule by HIC are used mainly by diabetics sufferers.



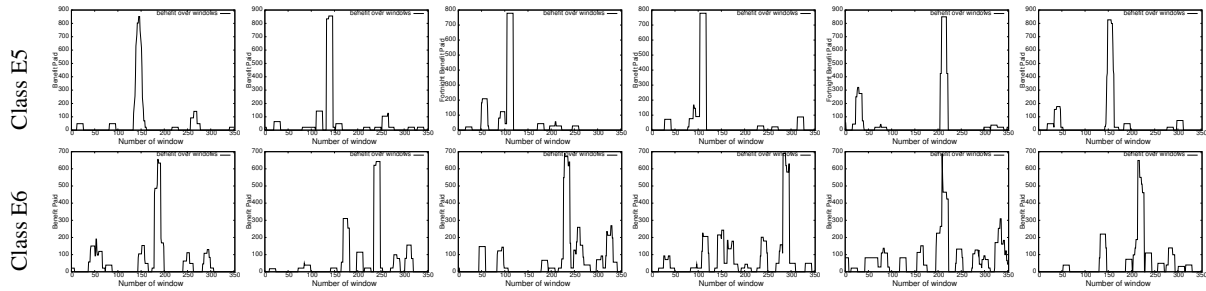
**Fig. 6.** Classification of diabetics (top-left), patients with chronic hepatitis (top-right), had a stroke (lower-left), and males (lower-right).

betic was classified. This means, statistically, any patients classified into these classes are highly unlikely suffering from diabetes. As a whole, the classification to diabetes provides a general picture about how likely an unknown patient is suffering from the disease once the profile has been classified by the HMMs.

## 5.2 Experiment 2: Chronic hepatitis

1,068 (out of 91,219) patients have been identified to suffer from chronic hepatitis <sup>11</sup>. The development of chronic hepatitis (hepatitis B or hepatitis C or viral hepatitis) can trigger diseases which need additional treatments. Thus, the classification shown in Figure 6 on the top-right can be explained similarly to that of the diabetes cases. When we compare the two graphs on diabetes and hepatitis respectively on the top row of Figure 6, the most revealing feature is that the highest class accommodates significantly more patients who were suffering from hepatitis than diabetes, i.e., a patient classified into class F3 is more likely to be suffering from hepatitis than diabetes. Note that there are about 4.5% patients suffering both chronic hepatitis and diabetes, thus, implies that this class collects patients whose health situation is complicated.

<sup>11</sup> Items 69432, 69435, 69447, 69453, 69456, 69272, 69273, 69275, 69277, 69278 as defined in the Medicare Benefits Schedule by HIC relate to treatment of hepatitis patients.



**Fig. 7.** Profiles of patients who suffer from diabetes, chronic hepatitis and stroke.

### 5.3 Experiment 3: Stroke

A set of patients who suffered from a stroke is considered; the classification of these patients is shown in Figure 6 on the lower-left. It is observed that these patients are found only in a very restricted set of classes. This shows that there are certain illnesses which require relatively specific courses of treatment, and hence, the claim patterns produced by affected patients are quite unique. Again the HMMs are capable of detecting these behavioral patterns.

### 5.4 Experiment 4: Male patient

Here we consider an extreme case where the feature of being a “male patient” is not supposed to have any significant impact on patient’s medical behavior<sup>12</sup> therefore it is not expected that this feature will cause differentiation among classes, i.e., we do not expect male patients to be classified into a few classes only, rather, it should spread evenly over all the classes. The result shown on the lower-right in Figure 6 confirms this intuition.

### 5.5 Summary

Figure 7 shows how patients having different properties can have similar medical patterns. The same token also explains why patients of similar property are classified into different classes. The first 2 profiles in each row corresponding to Class  $E_5$  and Class  $E_6$  are profiles of diabetics, the middle 2 are patients suffering from chronic hepatitis and last 2 are patients who had stroke. As indicated, by using the a priori information based on the course of treatment taken, it is possible to distinguish these patterns into ones which suffered from diabetes, hepatitis, or stroke respectively. Thus, even though the temporal behaviours are very similar, by using expert information about treatments which relate to a known disease we are able to disambiguate these ambiguous profiles into illness classes with a specific level of probability.

<sup>12</sup> This observation on male patients is typical for this cohort group as it is not expected to find many gender related medical treatments. However, it can be expected that a different observation is made in other age cohort groups. For example, in the age cohort group of 25 to 35 year old we would expect a very different result as female patients will claim for pregnancy related treatments while the males will not.

Once the system is fully trained, unseen profiles can be classified, and labeled with probability values which indicate the likelihood with which the patient is suffering from a specific illness. This automatic annotation process will be very useful in enabling the database to be used for other purposes, e.g., public health investigations.

## 6 Conclusions

In general, hidden Markov models are not particularly suitable for data mining applications as the computational expenses are prohibitive when sets of training data are large. We have developed an effective methodology to overcome this limitation. Our approach first decomposes the data set into groups of patients of similar age since the age contributes significantly to a patient's medical record properties. Then the approach applies recursively the Gaussian mixture clustering and HMMs procedures on randomly chosen samples from the training set until a convergence in the classification error is observed. This method is effective in detecting redundancies in the data set and hence can contribute significantly to the reduction of the computational effort of the HMMs. The experiments confirmed that with the help of the proposed methodology HMMs can be employed to data mining tasks. Secondly, by using some a priori information, e.g., item usage, we are able to automatically label profiles in the clusters into patients who might be suffering from various illnesses. While this classification may be coarse, as we do not have any diagnostic information. Nevertheless this coarse classification would enable the labelled database to be used for other investigations, thus adding values to the database.

## 7 Acknowledgement

The second and third authors wish to acknowledge financial support provided by the Australian Research Council through a SPIRT grant. They also acknowledge many supports provided by the Health Insurance Commission without which this research could not have been carried out.

## References

1. Juang, B. H., Levenson, S. E., Sondhi, M. M. "Maximum likelihood estimation for multivariate mixture observations of Markov chains", *IEEE Trans on Information Theory*. Vol 32, pp 307-309, 1986.
2. Liporace, L. A., "Maximum likelihood estimation for multivariate observations of Markov sources". *IEEE Trans on Information Theory*, Vol 28, pp 729-734, 1982.
3. McLachlan, G. J., Peel, D. "Finite Mixture Models", Wiley New York, 2000.
4. Duda, R.O., Hart, P.E., "Pattern recognition and scene analysis", J Wiley: New York, 1972.
5. Jeffrey, D., Banfield, Adrian E. Raftery, "Model-based Gaussian and non-Gaussian clustering", *Biometrics* Vol 49, pp 803-821, 1993.
6. Kohonen, T., "Self-Organizing Maps", Springer, Second Extended Edition 1997, 1995.
7. J.J.Verbeek, N.Vlassis, B. Krose. "Efficient Greedy Learning of Gaussian Mixture Model", *Neural Computation* Vol 15, Issue 2, pp 469-485, 2003
8. Herman J. Bierens. "Information criteria", <http://econ.la.psu.edu/hbierens/INFCRIT.PDF>.
9. Deller, J. R. Jr., Proakis, J. G., Hansen, J. H. L. "Discrete-time Processing of Speech Signals", MacMillan Publishing Company: New York, 1993.
10. Padhraic Smyth, "Clustering Sequences with Hidden Markov Models", *Advances in Neural Information Processing Systems*, Vol.9, pp 648-, The MIT Press, 1997.

# **A Data Mining Approach to Analyze the Effect of Cognitive Style and Subjective Emotion on the Accuracy of Intuitive Time-Series Forecasting**

Hung Kook Park<sup>1</sup>, Byoungho Song<sup>1</sup>, Hyeon-Joong Yoo<sup>2</sup>, Dae Woong Rhee<sup>1</sup>, Kang Ryoung Park<sup>1</sup> and Juno Chang<sup>1</sup>

<sup>1</sup>Division of Media Technology, College of Computer Software & Media Technology  
Sangmyung University

7 Hongji-dong, Jongno-gu, Seoul, Republic of Korea 110-743

[{parkh, bhsong, rheec219, parkgr, jchang}@smu.ac.kr](mailto:{parkh, bhsong, rheec219, parkgr, jchang}@smu.ac.kr)

<sup>2</sup>Division of Information Technology & Communication, College of Engineering  
Sangmyung University

San 98-20, Anseo-dong, Cheonan, Chungcheongnam-do, Republic of Korea 330-720

[yoohj@smu.ac.kr](mailto:yoohj@smu.ac.kr)

**Abstract.** Data mining is finding hidden rules in given dataset using non-traditional methods. The objective is to discover some useful tendency or patterns from the given collection of data. This research investigates if the differences in accuracy of “time series forecasting” are related to the differences in one’s cognitive style and subjective emotion. Two kinds of analyses were performed in advance of applying a data mining technology. Firstly, a statistical test was executed and the hypotheses established in the research model for the statistical test did not have the positive correlation between each of cognitive styles and the accuracy of intuitive time-series forecasting. Secondly, a self-organizing neural network (SONN) was utilized for analyzing the correlation and comparing the relative degree of correlation. The results showed a correlation but did not tell whether the correlation was a positive one or a negative one. Therefore, data mining approach was used to discover which positively influence intuitive forecasting. We have tried to find out any consistent tendencies in the frequent rules and found that there were positive correlations in some parts. Subjects in analytic style showed more accurate and the subjects in relax mode showed more accurate as well.

**Keywords:** data mining, neural network, self-organizing neural network, self-supervised adaptive neural network, cognitive style, emotion, intuitive forecasting, decision making

## 1 Introduction

The activities of judgemental time-series forecasting can be easily found in our real life such as estimating the movement of stock exchange indices, weather forecasting, and so on.

Through researches on intuitive judgement and cognitive styles, Kuo [1] discovered that the top economists rely on their keen intuition to aggressively solve their problems. Knowledge necessary for problem solving is dispersed in one's inmost thoughts and environs, which explains why intuition may be able to more effectively solve dynamic and abstract problems. In addition, most of businesses rely on intuitive forecasting as their main tools in their business activities as more experiments prove that "judgemental forecasting" is more accurate and efficient compared to statistical forecasting [2]. Ruble and Cosier [3] studied on effects of cognitive styles and decision setting on performance based on 162 economic-majoring students. Davis, Grove and Knowles [4] divided 96 graduate students into categories of four decision making styles and put them through computer simulation, which situated them in an economic environment. The result confirmed significant differences in cost effectiveness among different decision making styles. Furthermore, it was discovered that intuitive decision making was more likely to be used when there is high uncertainty, no past data or experience is available, many variables are scientifically unpredictable, there is time constraint, or many alternatives exist [5].

This research has executed many experiments to analyze the effect of decision maker's cognitive style and subjective emotion on the accuracy of intuitive time-series forecasting. Unfortunately the effect or meaningful relationship between them could not easily revealed by the traditional statistical analysis method. Then this research used the self-supervised adaptive algorithm [6] to find out any correlation between them. The results showed a correlation but did not tell whether the correlation was a positive one or a negative one. And then this research decided to apply the data mining [7, 8, 9, 10], a new approach to find something meaningful and hidden from the given collection of data, to our experimental measurements.

## 2 Research Methodology

### 2.1 Research Hypotheses

The study examined the correlation between cognitive styles and their final outcomes. The following hypothesis could be constructed:

H1: Accuracy in "time-series forecasting" differs among different subjective emotion.

H2: Accuracy in “time-series forecasting” differs among different cognitive styles.

## 2.2 Experimental Design

IT junior and senior undergraduate students were used as test subjects. The subjects have taken decision making related classes in the past. The researchers first evaluated cognitive styles of 29 students, and measured their forecasting error. Then 48 students were added to get enough number of students for each cognitive style. Hence, the total number of subjects was 77.

The experiment was on time-series forecasting. Time series data was driven from M-competition [11]. More precisely, the time-series data given to the test subjects was number of PCs sold in a month and they were to assume that they were PC sales managers. Total of forty data were given which was the sales volume for each month for the period of three years and four months. They were asked to predict the sales volume for next eight months. No other data such as cause-and-effect data were not provided except for the given time-series data.

In order to minimize variance in experiment, one person performed the entire test while standardizing the instruction given for all subjects.

Process of experiment was as follows:

- (1) Read the instruction when the subject enters the room.
- (2) The researcher gives a brief summary of the experiment.
- (3) Prior to the experiment, measure subject's subjective emotion.
- (4) Collect the subjective emotion survey.
- (5) Proceed with the test (app. 2 min.)
- (6) End the test.
- (7) Measure subject's cognitive style

## 2.3 Measure and Observation

**Independent Variables.** Independent variables are subject's subjective emotion and cognitive style.

**Subjective emotion:** The subjective emotion survey tool using five-point Lickert scale developed by the researchers was used to measure the subjective emotions such as i(negative-alert), ii(negative-relaxed), iii(positive-alert) and iv(positive-relaxed).

**Cognitive style:** This research adopted the decision style classification scheme, which is the basis for many measures of decision style including the popular Myers-Briggs Type Indicator test [12]. And this research used Alan Rowe's Decision Style Inventory to measure the subjects' decision styles such as A(analytic), B(behavioral), C(conceptual), and D(Directive) [13].

**Dependent Variable.** Accuracy of time-series forecasting is measured by the mean absolute percent error (MAPE). MAPE is a universally used tool in time-series forecasting and represented in absolute percentage value of standard deviation of forecasted value from actual value. The range of possible MAPE values is 0 to 1. The lower the MAPE value is, the better the accuracy should be.

### 3 Results of the Analyses by Statistical Test and Neural Network

#### 3.1 Statistical Hypothesis Testing

T test and ANOVA were used for the results of this experiment. For statistical package, SPSS for Windows was used.

Table 1 and 2 showed the results of the analysis which are as follows:

- (1) No differences exist in accuracy of time-series forecasting between different subjective emotions.
- (2) No differences exist in accuracy of time-series forecasting between different cognitive styles.

**Table 1.** ANOVA: Subjective Emotion vs MAPE

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	.014	3	.005	.556	.647
Within Groups	.373	44	.009		
Total	.392	47			

**Table 2.** ANOVA: Cognitive Style vs MAPE

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	.006	3	.002	.217	.884
Within Groups	.386	44	.009		
Total	.392	47			

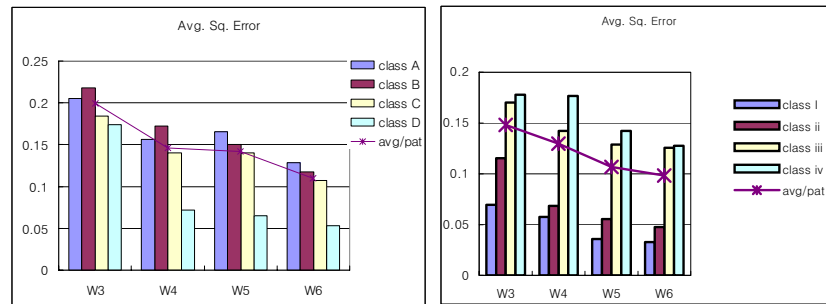
#### 3.2 Self-Organizing Neural Network

Since the effect or meaningful relationship between them could not be easily revealed by the traditional statistical analysis method, the researchers attempted to analyze the

experiment using a self-organizing neural network [14, 15, 16, 17] to determine the degree of correlation among the parameters. In this analysis, the self-supervised adaptive neural network (SSANN) proposed by Luttrell [6] was used. It can employ asymmetric neighborhood functions. And when there exists correlation between input vectors of different neuron clusters, the degree of asymmetry and the reconstruction error shows systematic relationships, e.g., the reconstruction error decreases with the increase of the asymmetry.

Fig. 1 shows the mean squared error after training the network for each style and each emotion. The bars show the result for each style, and the marked line curve shows their average over patterns. According to fig. 1, the results of the analysis [which] are as follows:

- (1) There exists correlation between cognitive style and time-series forecasting accuracy; subjective emotion and time-series forecasting accuracy.
- (2) The correlation between cognitive style and time-series forecasting accuracy is higher than that between subjective emotion and time-series forecasting accuracy.
- (3) Style C has higher correlation with time-series forecasting than A and B styles and B has higher correlation than A. It seems that emotion ii has higher correlation than emotion iii.



**Fig. 1.** The training results with the SSANN for each cognition style and subjective emotion.

Since there is no absolute reference data we used the property of the self-supervised adaptive neural network that inherently used the correlation between inputs, to figure out the existence of correlation and to compare correlation degrees. We found that there was a correlation between cognition characteristics and decision-making.

## 4 Data Mining Approach

As described in the previous section, two kinds of analyses were performed in advance of applying a data mining technology failed to tell whether the correlation was a positive one or a negative one. Therefore, data mining approach was used to discover which positively influence intuitive forecasting.

Data mining is finding hidden rules in given dataset using non-traditional methods [4]. The objective is to discover some useful tendency or patterns from the given collection of data. This research had mined the rules representing the effect of the cognitive style and the subjective emotion on the accuracy of the subjects' judgemental time-series forecasting, and then had tried to find out any consistent tendencies in the frequent rules.

Several techniques have been proposed for the actual data mining [18, 19]. In this research, the researchers used the "ROSSETA" which is a data mining tool for MS Windows developed by the Department of Computer and Information Science in Norwegian University of Science and Technology in 1999 [19].

### 4.1 Preparation for Data Mining

For each subject, we have 1 style value (A, B, C, or D), 4 (original) subjective emotion values (in numeric) and 1 MAPE value (in numeric).

In general data mining requires partitioning every continuous (numeric) value range into several zones to find the tendency because it is difficult to measure the frequency of each value in the continuous value domain due to the numerous numbers of the different values. The researchers partitioned all the numeric properties into three levels: high, low, and middle. Highest 30% was assigned to 'high'; Lowest 30% was assigned to 'low'; and the rest 40% was assigned to 'middle'. Thus 23 of 77 MAPE values were 'high', 31 of 77 MAPE values were 'middle', and 23 of 77 MAPE values were said to be 'low'. The same values at any boundary were considered to be 'middle'.

### 4.2 Mining Results

The researchers have found so many rules (relationships) between arbitrary pair of properties. Infrequent rules were removed and tried to find out any consistent tendencies in the rest frequent rules. The rest of this section consists of the observations. The portions of high accuracy (low MAPE), middle accuracy (middle MAPE), and low accuracy (high MAPE) will be written in this order in '(' and ')' at the end of any tendencies or rules.

#### Observations on the Effect of Cognitive Style

*Observation 1:* The subjects in style A had a tendency to make high accurate (low MAPE) forecasting (10/25, 9/25, 6/25).

*Observation 2:* The subjects in style B had a tendency to make low accurate forecasting (3/17, 8/17, 6/17).

And we can't find any meaningful tendencies in style C (7/23, 9/23, 7/23) and style D (3/12, 5/12, 4/12).

### **Observations on the Effect of Subjective Emotion**

*Observation 3:* There was a tendency that regardless of positive or negative emotion, the higher relaxed level the subject shows at forecasting, the higher accuracy (s)he achieves, and when the lower relaxed level is shown, the lower accuracy is achieved. The evidence is:

negative-relaxed (low)  
-> (4/11, 1/11, 6/11)  
positive-relaxed (mid)  
-> (5/22, 9/22, 8/22)  
positive-relaxed (high)  
-> (6/12, 3/12, 3/12)

*Observation 4 :* In contrast, there was a tendency that regardless of positive or negative emotion, the higher alert level the subject shows at forecasting, the lower accuracy (s)he achieves, and when the lower alert level is shown, the higher accuracy is achieved.

The evidence is :

negative-alert (high)  
-> (3/12, 4/12, 5/12)  
positive-alert (low)  
-> (5/11, 4/11, 2/11)  
positive-alert (high)  
-> (2/10, 3/10, 5/10)

### **4.3 Analysis of Results**

It is found that there are positive correlations in some parts. Subjects in analytic style showed more accurate and the subjects in relax mode showed more accurate as well.

Subjects in style A (Analytic) seem to be more accurate, and subjects in style B (Behavioral) seem to be less accurate. It means that if we hire analysts in style A, we would have more opportunity to be happy.

Subjects in relaxed mode seem to be more accurate. It means that if we make our analysts relaxed, we would have more opportunity to be happy.

## 5 Concluding Remarks

This research analyzed in various ways by using cognitive style data and subjective emotion data to discover which positively influence intuitive forecasting.

In advance of applying a data mining technology, a statistical test was executed but the results of the statistical test did not show the positive correlation between each of cognitive styles and the accuracy of intuitive time-series forecasting. And then a self-organizing neural network (SONN) was utilized for analyzing the correlation and comparing the relative degree of correlation. The results showed a correlation but did not tell whether the correlation was a positive one or a negative one. However in the data mining approach, positive correlations were found in some parts. Table 3 summarizes the comparison of three analyses findings.

In conclusion, the data mining approach discovered the more meaningful relationship between the accuracy of time-series forecasting and both the cognitive style and the subjective emotion than the statistical test and the neural network approach.

**Table 3.** Comparison of three analyses findings

	Statistical Test	MAPE	
		SONN	Data Mining
Subjective Emotion	no significant difference	correlated	positively correlated in parts
Cognitive Style	no significant difference	correlated	positively correlated in parts

The limitation of this research is that the uncontrolled external variable during the experiment might cause the lack of correlation in the statistical analysis and there might be relatively not enough data for the neural network to analyze the correlation between the accuracy of time-series forecasting and both the cognitive style and the subjective emotion.

## References

1. Kuo, F. "Managerial intuition and the development of executive support systems," *Decision Support Systems*, 24 (1998) 89-103.

2. Lim, J., Whang, M., Park, H., Lee, H.: "A physiological approach to the effect of emotion on time series judgemental forecasting: EEG and GSR," *Korean Journal of the Science of Emotion and Sensibility*, Vol. 1, No. 1 (1998) 123-133
3. Ruble, T., Cosier, R.: "Effects of cognitive styles and decision setting on performance," *Organizational Behavior and Human Decision Process*, Vol. 46, No. 2 (1990) 283-312
4. Davis, D., Grove, S., Knowles, P.: "An experimental application of personality type as an analogue for decision-making style," *Psychological Report*, 66-1 (1990) 167-184
5. Agor, W.: "The logic of intuition: How top executives make important decisions," *Organizational Dynamics*, 14-3 (1986) 5-23.
6. Luttrell S. P.: "Self-supervised adaptive networks", *IEEE Proceedings-F*, vol. 139 (1992) 371-377
7. Agrawal, R., Imielinski, T. and Swami, A. "Database Mining: A Performance Perspective," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 5, No. 6, Dec. 1993.
8. Barson, A. and Smith, S. J. *Data Warehousing, Data Mining, and OLAP*, McGraw-Hill Pub., 1997.
9. Chen, M.-S. Han, J. and Yu, P. S. "Data Mining: An Overview from a Database Perspective," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 8, No. 6, Dec. 1996, pp. 866-883.
10. Frawley, W. J., Piatetsky-Shapiro, G. and Matheus, C. J. "Knowledge Discovery in Databases: An Overview," *Knowledge Discovery in Databases*, AAAI/MIT Press, 1991, pp. 1-27.
11. Makridakis, S.: *Forecasting Competition*, <http://www.insead.fr/Research/ForecastCompet>.
12. Myers, I.: *Manual for the Myers-Biggs Type Indicator*, Princeton, NJ: Educational Testing Service (1962)
13. Rowe, A., Boulgarides. *Managerial Decision Making*. Englewood, Clifff, NJ: Prentice Hall (1994)
14. Kohonen, T: *Self-Organizing Maps*, Springer (1995)
15. Srivastava, L. , Singh, S.N. , Sharma, J.: "Estimation of loadability margin using parallel self-organizing hierarchical neural network" *Computers & electrical engineering*, v.26 no.2 (2000) 151-167
16. Oh, S., Pedrycz, W.: "The design of self-organizing Polynomial Neural Networks" *Information Sciences*, v.141 no.3/4 (2002) 237-258
17. Oh, S., Pedrycz, W., Ahn, T.: "Self-organizing polynomial neural networks based on polynomial and fuzzy polynomial neurons: analysis and design," *Fuzzy sets and systems*, v.142 no.2 (2004) 163-198
18. Kdnuggets, "Software for Data Mining and Knowledge Discovery," <http://www.kdnuggets.com/software/index.html>, (May 3, 2000).
19. Knowledge Systems Group, Dept. of Computer and Information Science, Norwegian University of Science and Technology, Trondheim, Norway, "The ROSSETA Homepage," <http://www.idt.unit.no/~aleks/rosetta/rosetta.html> (May 3, 2000).



# Decision Models for Record Linkage

Lifang Gu<sup>1</sup> and Rohan Baxter<sup>2</sup>

<sup>1</sup> CSIRO ICT Centre  
GPO Box 664, Canberra, ACT 2601, Australia  
[Lifang.Gu@csiro.au](mailto:Lifang.Gu@csiro.au)

<sup>2</sup> Australian Taxation Office  
51 Allara Street, Civic, Canberra, ACT 2601, Australia  
[Rohan.Baxter@ato.gov.au](mailto:Rohan.Baxter@ato.gov.au)

## Abstract.

The process of identifying record pairs that represent the same real-world entity in multiple databases, commonly known as record linkage, is one of the important steps in many data mining applications. In this paper, we address one of the sub-tasks in record linkage, i.e., the problem of assigning record pairs with an appropriate matching status. Techniques for solving this problem are referred to as decision models. Most existing decision models rely on good training data, which is, however, not commonly available in real-world applications. Decision models based on unsupervised machine learning techniques have recently been proposed. One such model is based on clustering. However, such clustering-based decision models tend to generate a large proportion of record pairs for costly clerical review. In this paper, we review several existing decision models and then propose an enhancement to such cluster-based decision models. The enhanced model first clusters all record pairs into matches and non-matches. A refinement step, which is the core of our enhancement, is then applied to identify record pairs for clerical review using a distance-based metric. Experimental results show that our proposed decision model achieves the same accuracy of existing models with a much smaller number of record pairs required for manual review. The proposed model also provides a mechanism to trade off the accuracy with the number of record pairs required for clerical review.

**Keywords:** data linking, record linkage, probabilistic linking, decision model, clustering, classification.

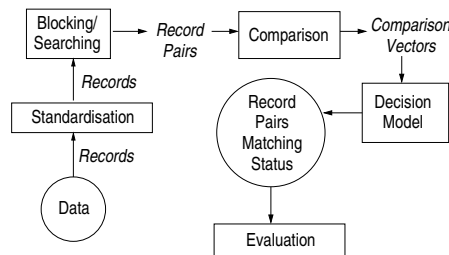
## 1 Introduction

Record linkage is the task of identifying records corresponding to the same entity from one or more data sources. Entities of interest include individuals, companies, geographic regions, families, or households. Record linkage has applications in systems for marketing, customer relationship management, fraud detection, data warehousing, law enforcement and government administration.

In many data mining projects it is often necessary to collate information about an entity from more than one data source. If a unique identifier is available,

conventional *SQL* ‘join’ operations in database systems can be used for record linkage, which assumes error-free identifying fields and links records that match exactly on these identifying fields. However, real-world data is ‘dirty’ and sources of variation in identifying fields include lack of a uniform format, changes over time, misspellings, abbreviations, and typographical errors.

Record linkage can be considered as part of the *data cleaning* process, which is a crucial first step in the knowledge discovery process [1]. Fellegi and Sunter [2] were the first to introduce a formal mathematical foundation for record linkage. Their original model has since been extended and enhanced by Winkler [3].



**Fig. 1.** Information flow diagram of a record linkage system

No matter what technique is used, a number of issues need to be addressed when linking data. Figure 1 shows the information flow diagram of a typical record linkage system as implemented in *TAILOR* [4] and *Febrl* [5].

Often, data is recorded or captured in various formats, and data fields may be missing or contain errors. Standardisation is an essential first step in every linkage process to clean and standardise the data. Since potentially every record in one dataset has to be compared with every record in a second data set, *blocking* or *searching* techniques are often used to reduce the number of comparisons. These techniques use *blocking variables* to group similar records together and therefore partition the data sets into smaller blocks (clusters). Only records within the same block are then compared in detail using the defined *comparison variables* and *functions*. The comparison vectors generated by such detailed comparison functions are passed to the decision model to determine the final status (match, non-match or possible match) of record pairs. The results of the record linkage can be assessed by the evaluation model.

The main challenges in record linkage are computational complexity and linkage accuracy. Recent developments in information retrieval, database systems, machine learning and data mining have led to improvement in the efficiency and accuracy of record linkage systems [6, 7].

In this paper, we focus on the decision model component of a record linkage system. The linkage accuracy in a record linkage system depends heavily on the decision model. We review several existing decision models and identify problems with these models. An enhanced clustering-based decision model is proposed.

The main contribution of this paper is the development of an enhanced clustering-based decision model as well as the introduction of some performance metrics. The key feature of our proposed decision model, compared to other existing models [4, 8], is that clustering is initially performed based on two clusters (matched and unmatched). A refinement step is then applied to identify record pairs with an uncertain matching status by using a metric introduced in this paper. The enhancement step also provides a mechanism to trade off the linkage accuracy with the amount of clerical review work.

The rest of the paper is organised as follows. Problems and notations are introduced in Section 2. Several existing decision models are reviewed and their limitations are identified in Section 3. In Section 4 we then present our enhanced decision model for addressing some of the identified limitations. Experimental results are described in Section 5 and conclusions are made in Section 6.

## 2 Definition, Notation, and Problem

For two data sources  $A$  and  $B$ , the set of ordered record pairs  $A \times B = \{(a, b) : a \in A, b \in B\}$  is the union of two disjoint sets,  $M$  where  $a = b$  and  $U$  where  $a \neq b$ . The former set,  $M$ , is usually referred to as *matched* and the latter set,  $U$ , as *unmatched*. The problem is to determine which set each record pair belongs to. In practice, a third set  $P$ , *possibly matched*, is often introduced to accommodate situations where the matching status of a record pair cannot be decided with information available from the data sources. If a record pair is assigned to  $P$ , a domain expert must manually examine the pair. Here we assume that a domain expert can always identify the correct matching status ( $M$  or  $U$ ) of such a record pair with or without extra information.

Assume that  $n$  common attributes,  $f_1, f_2, \dots, f_n$ , of each record from sources  $A$  and  $B$  are chosen for comparison. For each record pair  $r_{i,j} = (r_i, r_j)$ , the attribute-wise comparison results in a vector of  $n$  values,  $c_{i,j} = [c_1^{i,j}, c_2^{i,j}, \dots, c_n^{i,j}]$  where  $c_k^{i,j} = C_k(r_i.f_k, r_j.f_k)$  and  $C_k$  is the comparison function that compares the values of the record attribute  $f_k$ . The vector,  $c_{i,j}$ , is called a *comparison vector* and the set of all the comparison vectors is called the *comparison space*. A comparison function  $C_k$  is a mapping from the Cartesian product of the domain(s),  $D_k$ , for the attribute  $f_k$  to a comparison domain  $R_k$ ; formally,  $C_k : D_k \times D_k \rightarrow R_k$ . One example of a simple comparison function is

$$C_I(v_1, v_2) = \begin{cases} 0 & \text{if } v_1 = v_2 \\ 1 & \text{otherwise} \end{cases} \quad (1)$$

where  $R_I = \{0, 1\}$ . The value computed by  $C_I$  is called a *binary comparison value*. Two additional types of comparison values produced by comparison functions are *categorical* and *continuous*.

The role of a decision model is to determine the matching status of a record pair given its comparison vector  $c_{i,j}$ . Depending on the type of comparison values and whether training data is needed, decision models of varying complexity have

been proposed in the literature. In the following section, we review some of these existing models.

### 3 Existing Decision Models

In this section, we review four existing decision models. The probabilistic model described in Section 3.1 was developed in 1969 and is still widely used in the medical and statistical domains. The other three models have been proposed recently to address some of its limitations.

#### 3.1 Error-Based Probabilistic Model

The probabilistic model defined by Fellegi and Sunter [2] assigns a weight  $w_k^{i,j}$  for each component of a record pair, i.e.,

$$w_k^{i,j} = \begin{cases} \log(m_k/u_k) & \text{if } c_k^{i,j} = 0 \\ \log((1 - m_k)/(1 - u_k)) & \text{if } c_k^{i,j} = 1 \end{cases} \quad (2)$$

where  $m_k$  and  $u_k$  are the probabilities of observing that the two values of the attribute  $k$  are the same for record pair  $r_{i,j}$  among matched and unmatched record pairs respectively. Mathematically, they are defined as:

$$\begin{aligned} m_k &= \text{Prob}\{c_k^{i,j} = 0 | r_{i,j} \in M\} \\ u_k &= \text{Prob}\{c_k^{i,j} = 0 | r_{i,j} \in U\} \end{aligned} \quad (3)$$

It can be seen that the weight  $w_k^{i,j}$  is large (positive) for a matched pair and small (negative) for an unmatched pair. A decision is made for each record pair by calculating a composite weight  $L(r_{i,j}) = \sum_{k=1}^n w_k^{i,j}$ , and comparing this value against two threshold values  $t_1$  and  $t_2$  where  $t_1 < t_2$ . Specifically, the decision is made as follows:

$$\begin{aligned} r_{i,j} &\in M \text{ if } L(r_{i,j}) \geq t_2 \\ r_{i,j} &\in U \text{ if } L(r_{i,j}) \leq t_1 \\ r_{i,j} &\in P \text{ if } t_1 < L(r_{i,j}) < t_2 \end{aligned} \quad (4)$$

The main issue in this model is therefore to determine estimates of the conditional probabilities  $m_k$  and  $u_k$  for  $k = 1, 2, \dots, n$ , as well as estimates of the two thresholds  $t_1$  and  $t_2$ . Two methods for estimating the conditional probabilities  $m_k$  and  $u_k$  were proposed by Fellegi and Sunter [2]. Winkler [9] uses the EM (Expectation Maximisation) method to estimate these conditional probabilities. However, all these methods rely on training data to estimate the parameters. The EM approach proves to be more stable and less sensitive to initial values.

#### 3.2 Cost-Based Probabilistic Model

In the above error-based probabilistic model, the thresholds  $t_1$  and  $t_2$  are estimated by minimising the probability of the error of making an incorrect decision

for the matching status of a record pair. This implicitly assumes that all errors are equally costly. However, this is rarely the case in many applications. Therefore, the minimisation of the probability of the error is not the best criterion to use in designing a decision rule because misclassification of different record pairs may have different consequences.

Verykios et al. [8] propose a decision model that minimises the cost of making a decision. Specifically, they use a constant error cost Bayesian model to derive the decision rule for a given cost matrix. For record linkage, the cost matrix  $D$  is  $3 \times 2$  in dimension. Let us denote by  $d_{ij}$  the cost of making a decision  $i$  when the record pair to be compared corresponds to one with an actual matching status  $j$ . Here  $i$  corresponds to one of the three regions decided by a decision rule in the decision space, namely matched  $M$ , possibly matched  $P$ , and unmatched  $U$  respectively, while  $j$  refers to the actual matching status  $M'$  and  $U'$ . The decision rule is obtained by minimising the mean cost  $\bar{d}$ , which is written as follows:

$$\begin{aligned} \bar{d} = & d_{MM'} \cdot \text{Prob}(M, M') + d_{MU'} \cdot \text{Prob}(M, U') + \\ & d_{PM'} \cdot \text{Prob}(P, M') + d_{PU'} \cdot \text{Prob}(P, U') + \\ & d_{UM'} \cdot \text{Prob}(U, M') + d_{UU'} \cdot \text{Prob}(U, U') \end{aligned} \quad (5)$$

where  $\text{Prob}(i, j)$  denotes the joint probability that a decision  $i$  is taken when the actual matching status is  $j$ . By using the Bayes theorem and replacing the above probabilities with the *a priori* probabilities of  $M'$  and  $U'$ , and the probability densities of the comparison vectors given the matching status, the above equation can be summarised by a decision rule similar to that of the error-based model described in Section 3.1. The only difference is that the threshold values also depend on the cost matrix (see [8] for details).

### 3.3 Inductive Learning-Based Decision Model

One of the limitations of the above probabilistic models is that they can only handle binary or categorical comparison vectors. Decision models based on machine learning techniques can overcome this shortcoming. One such decision model is based on inductive learning techniques and can handle all types of comparison vectors [4].

In inductive learning, a training set of patterns, in which the class of each pattern is known *a priori*, is used to build a model that can be used afterwards to predict the class of each unclassified pattern. A training instance has the form of  $\langle x, f(x) \rangle$  where  $x$  is a pattern and  $f(x)$  is a discrete-value function that represents the class of the pattern  $x$ , i.e.,  $f(x) \in L_1, L_2, \dots, L_l$  where  $l$  is the number of the possible classes. In the case of record linkage,  $x$  is the comparison vector  $c$ ,  $l$  is 2 ( $M$  and  $U$ ), and  $f(c)$  is the corresponding matching status, i.e.,  $f(c) \in M, U$ . One of the popular classification techniques is decision trees, which exploit the regularities among observations in the training data. Predictions are made on the basis of similar, previously encountered situations. The accuracy of this type of decision model depends on the representativeness of the training data.

### 3.4 Clustering-Based Decision Model

A problem with inductive learning-based decision models, as well as with probabilistic decision models, is that they all rely on the existence of a training data set. However, training data is not usually available for most real-world applications. Therefore, unsupervised learning methods, such as clustering, have been introduced to the record linkage community since they do not require training data. Elfeky et al. [4] used the *k-means clustering* to group record pairs into three clusters: *matched*, *unmatched*, and *possibly matched*. They also described how to determine the matching status of each cluster once the clustering is completed.

However, the *possibly matched* record pairs do not necessarily form a distinctive cluster in real applications. It is usually assumed that the distribution of comparison values is bimodal. The 3-cluster *k-means* algorithm leads to a large cluster of the *possibly matched* record pairs as reported in [4]. This is undesirable for most real-world applications as clerical review is very costly.

## 4 Enhanced Clustering-Based Decision Model

In Section 3 we have identified limitations with existing decision models. These include the restriction to categorical comparison values, the need for training data sets and the large proportion of record pairs required for clerical review. In this section, we propose an enhanced clustering-based decision model that does not have these limitations.

Based on the observation that record pairs usually form two main clusters in the comparison space, the proposed model uses a clustering algorithm to partition the record pairs into matched and unmatched clusters initially. A third cluster is then formed by record pairs in a fuzzy region between the two main clusters. The matching status of these record pairs in the fuzzy region cannot be determined from the available information and therefore have to be resolved by a domain expert. We refer to this third cluster as the *possibly matched*. We introduce a distance-based metric used for identifying the fuzzy region. The size of the fuzzy region, which can be easily controlled by tuning a threshold parameter, determines the balance between the linkage accuracy and the amount of clerical review work.

### 4.1 Clustering Algorithms

There are many clustering algorithms [10, 11] available. The most widely used is the *k-means* clustering [10] because of its easy implementation and computational efficiency when *k* is small. The *k-means* algorithm is summarised as follows:

1. Partition the whole dataset into *k* clusters (the data points are randomly assigned to the clusters). This results in clusters that have roughly the same number of data points.
2. Compute the mean (centroid) of each cluster.

3. For each data point, calculate the distance from the data point to each cluster. If the data point is closest to its own cluster, leave it where it is. If it is not closest to its own cluster, move it into the closest cluster.
4. Repeat Steps 2 and 3 until no data point moves from one cluster to another.

Good results can be achieved by the  $k$ -means clustering algorithm if all points are distributed around  $k$  well separated clusters. The shape of these  $k$  clusters depends on the distance measure used. For example, if the Euclidean distance metric is used, the shape of the clusters is spherical for 3-dimensional data.

For our decision model, other clustering algorithms, such as model-based clustering [11], can also be used.

## 4.2 Fuzzy Region Identification

When the initial clustering process is completed, all record pairs are assigned to one of the two main clusters. However, there is usually a grey or fuzzy region where record pairs of true matches and non-matches co-exist. Here we introduce a metric to identify this fuzzy region.

For  $k$ -means clustering, the distances of each point to the two cluster centres can be calculated. We denote the distances of point  $i$  to the two cluster centres by  $d_{i,1}$  and  $d_{i,2}$  respectively. Any distance metric, such as Euclidean or Mahalanobis, can be used. To identify the fuzzy region, we define  $\Delta d_i$ , the relative distance difference of point  $i$  to the two cluster centres, as follows:

$$\Delta d_i = \frac{|d_{i,1} - d_{i,2}|}{(d_{i,1} + d_{i,2})/2} \quad (6)$$

where the denominator is the average of  $d_{i,1}$  and  $d_{i,2}$ . If  $\Delta d_i$  is small, point  $i$  has approximately same distances to the two cluster centres. Therefore, points with small  $\Delta d_i$  values cannot be assigned to one of the two clusters with certainty and they form the fuzzy region. The size of this fuzzy region can be controlled by a parameter, the threshold  $T_d$ , the maximum acceptable relative distance difference. The value of the threshold  $T_d$  can be determined based on available resources for manual review and the required accuracy. All points with a  $\Delta d$  value smaller than  $T_d$  would be then assigned to the *possibly matched* cluster.

For other clustering algorithms, similar metrics can be defined. For example, the difference of the probabilities of each point belonging to each of the two clusters can be such a metric for model-based clustering.

Our enhanced clustering algorithm can therefore be considered as a normal 2-cluster clustering with an additional refinement step. During this refinement step record pairs in the fuzzy region are reassigned to the *possibly matched* cluster, based on their values of  $\Delta d_i$  and the given threshold value,  $T_d$ . As it will be shown in our experiment (Section 5), this provides an effective way of controlling the trade-off between the required linkage accuracy and the proportion of record pairs needed for clerical review.

## 5 Experimental Results

To evaluate the performance of our clustering-based decision model and compare it to existing decision models, an empirical experiment has been conducted.

### 5.1 Data Sets and Parameters

We use the database generator, *DBGen*, distributed as a part of Febrl package [5], to generate test data sets for our experiment. This tool can generate data sets that contain attributes, such as names, addresses, dates etc, based on various frequency tables. It generates *duplicates* of the original records by randomly introducing various modifications, the degrees of which are specified by the corresponding probabilities. We generate 4 data sets and their characteristics are shown in Table 1. Data set 1 contains 500 original records, each of which has a corresponding duplicate. In data set 2, the number of records is increased and also an original record can have a maximum of 5 duplicates. In data set 3, the number of duplicated records is larger than that of the original records. The duplicates of data set 4 are generated by doubling the default modification probabilities used in data set 2. Therefore, the difference between an original record and its duplicates in data set 4 is larger compared to those in data set 2. Note the number of true matched record pairs (column 5) is larger than the number of duplicates (column 3) for data sets 2, 3 and 4 because of the transitivity of the multiple duplicates.

dataset name	#original records	#duplicate records	#max dups per record	#total true matches	#pairs from blocking	#matched pairs
dataset 1	500	500	1	500	693	459
dataset 2	1,000	1,000	5	2,290	2,782	1,940
dataset 3	2,000	3,000	5	6,924	10,666	5,905
dataset 4	1,000	1,000	5	2,338	2,539	1,639

**Table 1.** Characteristics of test data sets generated by the Febrl database generator.

We applied the 3-pass standard blocking, i.e. 3 rounds of grouping record pairs based on 3 different blocking variables, on all four data sets. The number of record pairs generated by this blocking method for each data set is shown in column 6 of Table 1. The last column of Table 1 shows the number of true matched record pairs among the record pairs generated by blocking. It can be seen that blocking has efficiently reduced the number of record pair comparisons but also missed some true matched record pairs. In this paper, we do not compare the performance of blocking methods (see [12] for details).

Table 2 shows the comparison variables and the corresponding comparison functions used in our experiment. Most of these comparison functions return binary values except the approximate string comparator, which returns continuous comparison values in the range of  $[0.0, 1.0]$ . In Febrl, a binary comparison value

Comparison Variable	Comparison Function
given name	NYSIIS Encoding String Comparator
surname	Winkler Approximate String Comparator
wayfare name	Winkler Approximate String Comparator
locality name	Key Difference Comparator
postcode	Distance Comparator
age	Age Comparator

**Table 2.** Comparison variables and functions used in the experiment.

is converted to the weight using Equation 2 whereas a continuous comparison value is converted to the weight using the following equation:

$$w_k^{i,j} = \begin{cases} \log \frac{m_k}{u_k} - \frac{c_k^{i,j}}{c_{max}} (\log \frac{m_k}{u_k} + |\log \frac{1-m_k}{1-u_k}|) & \text{if } 0 \leq c_k^{i,j} \leq c_{max} \\ \log \frac{1-m_k}{1-u_k} & \text{if } c_k^{i,j} > c_{max} \end{cases} \quad (7)$$

where  $c_{max}$  is the maximum approximate string difference value tolerated. The disagreement weight is obtained when the comparison value exceeds  $c_{max}$  whereas a (partial) agreement weight is resulted for a smaller comparison value.

In our experiment, we compare our clustering-based model to other existing decision models, specifically to the probabilistic decision model implemented in Febrl [5] and the 3-cluster  $k$ -means model [4]. Since the probabilistic decision model takes the sum of all weights as input, clustering is also performed on this one dimensional feature. Note that clustering on individual comparison vector components can be easily performed and details are discussed in Section 5.3. All the parameters for the probabilistic decision model are set manually. Specifically, the conditional probabilities  $m$  (0.95) and  $u$  (0.01), and the maximum string difference value tolerated  $c_{max}$  (0.3) are fixed and the threshold values  $t_1$  and  $t_2$  vary within a certain range. For clustering-based decision models, we use the  $k$ -means clustering algorithm implemented in R [13] and  $k$  is equal to 2 and 3 for our record linkage application. The distance threshold value,  $T_d$ , used for controlling the size of fuzzy region varies from 0.1 to 1.0.

## 5.2 Performance Metrics

To compare different decision models, we need some performance metrics. Here we adopt two metrics proposed in [4] and the *recall* metric commonly used in information retrieval to evaluate the decision models.

Let  $N$  be the total number of record pairs generated by a blocking method, and  $n_{a,b}$  be the number of record pairs whose predicted matching status is  $a$ , and whose actual matching status is  $b$ , where  $a$  is either  $M$ ,  $U$  or  $P$ , and  $b$  is either  $M'$  or  $U'$ . For evaluation purposes, we assume that record pairs with a  $P$  status can be always correctly classified. The three metrics are defined as follows:

- *AC*: the *accuracy* metric, *AC*, tests how accurate a decision model is. It is defined as the proportion of the correctly classified (both matched and

unmatched) record pairs:

$$AC = \frac{n_{M,M'} + n_{U,U'} + n_{P,M'} + n_{P,U'}}{N} \quad (8)$$

- *PP*: the *PP* metric measures the proportion of the record pairs that are classified as *possibly matched* by a decision model, for clerical review:

$$PP = \frac{n_{P,M'} + n_{P,U'}}{N} \quad (9)$$

- *recall*: the *AC* metric does not distinguish accuracy between the matched and unmatched record pairs since it reflects the total classification accuracy. The original *recall* metric in information retrieval measures the number of relevant documents retrieved as fraction of all relevant documents. Here we use it to measure the accuracy of the decision model for matched record pairs and it is defined as the proportion of all matched record pairs that are classified correctly:

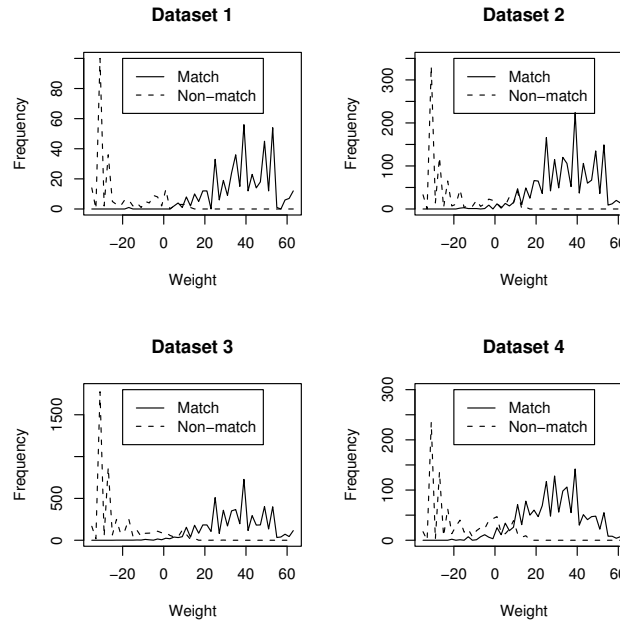
$$recall = \frac{n_{M,M'} + n_{P,M'}}{n_{M,M'} + n_{P,M'} + n_{U,M'}} \quad (10)$$

### 5.3 Results

Figure 2 shows distributions of the sum of weights (calculated by Equations 2 and 7) among the true matched and unmatched record pairs for our test data sets. It can be seen that the two clusters are separated reasonably well except for data set 4. The two clusters overlap in the middle with weight values ranging between  $-15$  and  $15$  for data set 4.

We have run the probabilistic decision model on the weights of these data sets and the results under the threshold values  $t_1 = 0$  and  $t_2 = 10$  are shown in Table 3. It can be seen that data set 4 has the lowest *AC* and *recall* values, and the highest *PP* value due to larger errors in the duplicate records. Figure 3 shows the *AC* and *PP* values of the probabilistic decision model for data set 4 under different threshold values. It can be seen that the linkage accuracy (*AC*) increases as  $t_1$  decreases and  $t_2$  increases. But this also leads to an increase in the proportion of record pairs (*PP*) for manual review. It is also evident that the increase in *AC* and *PP* values is greater when  $t_2$  increases, compared to the same amount of decrease in  $t_1$ . Similar trends have been observed for the other 3 data sets.

We have also run the *k*-means clustering algorithm on the same data sets and the results for two- and three-cluster models are also shown in Table 3. Clustering using 3-clusters have resulted in about 23% to 31% of record pairs being classified as possible matches. This is obviously impractical for most real world applications as they often involve large data sets. On the other hand, the accuracy values for the two-cluster case range from 0.919 to 0.980. Such accuracy might be acceptable for some applications, considering that this is done fully automatically and no manual review is required.



**Fig. 2.** Weight distributions of true matched and unmatched record pairs of the four test data sets.

If resources for manual review are available, a third cluster can be created by applying our proposed process of identifying record pairs in the fuzzy region for clerical review. Table 4 shows the results of our enhanced model for the four test data sets under different  $T_d$  threshold values. It can be seen that the accuracy for data set 1 increases from 0.980 to 0.999 by assigning about 7% of record pairs for clerical review. Similarly, the accuracy (or *recall*) for data set 4 also increases from 0.919 to 0.992 by assigning 27% of record pairs for manual review. This provides an effective mechanism to trade-off the *accuracy* and *recall* metrics with the number of record pairs for manual review. In practice, a  $T_d$  value in the range of 0.2 to 0.5 is a good starting point, depending on the quality of data sets and available resources for manual review.

Comparison of Table 3 with Table 4 shows that our enhanced decision model achieves the same accuracy/recall of the 3-cluster clustering model [4] with a much smaller  $PP$  value. For example, to achieve an  $AC$  value of 0.997 and a *recall* value of 0.998 for data set 1, our model assigns only 2.9% of record pairs needed for manual review while the existing 3-cluster clustering model allocates 29.7% of record pairs for manual review. This reinforces our observation that there is not always a distinctive cluster between the matched and unmatched clusters. The existing 3-cluster decision model therefore leads to a large number

Name	Decision Model								
	Probabilistic			Clustering ( $k = 2$ )			Clustering ( $k = 3$ )		
	AC	PP	recall	AC	PP	recall	AC	PP	recall
dataset 1	0.989	0.038	0.998	0.980	0.0	0.985	0.981	0.297	0.998
dataset 2	0.973	0.039	0.991	0.955	0.0	0.975	0.997	0.311	0.996
dataset 3	0.982	0.046	0.993	0.963	0.0	0.979	0.998	0.230	0.996
dataset 4	0.961	0.093	0.977	0.919	0.0	0.926	0.996	0.310	0.993

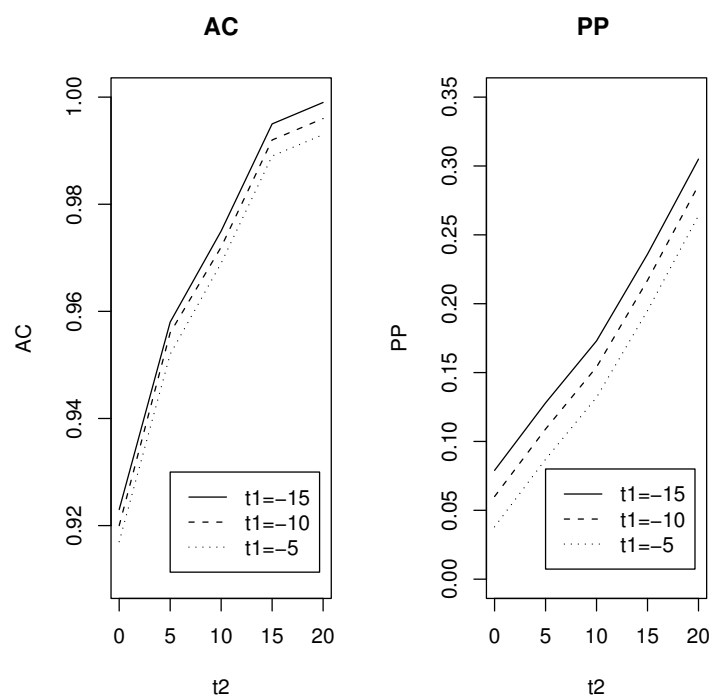
**Table 3.** Results of the probabilistic model for  $t_1 = 0$  and  $t_2 = 10$  and of the  $k$ -means clustering models for  $k = 2$  and 3.

$T_d$	Dataset											
	Dataset 1			Dataset 2			Dataset 3			Dataset 4		
	AC	PP	recall	AC	PP	recall	AC	PP	recall	AC	PP	recall
0.10	0.993	0.010	0.989	0.964	0.016	0.980	0.969	0.011	0.984	0.931	0.017	0.936
0.20	0.997	0.029	0.998	0.969	0.025	0.984	0.977	0.025	0.987	0.946	0.042	0.953
0.50	0.999	0.071	0.998	0.992	0.078	0.991	0.993	0.077	0.993	0.974	0.113	0.970
0.80	0.999	0.123	0.998	0.997	0.127	0.996	0.997	0.125	0.996	0.988	0.206	0.982
1.00	0.999	0.144	0.998	0.997	0.173	0.996	0.998	0.162	0.997	0.992	0.270	0.988

**Table 4.** Results of our enhanced decision model under various threshold values.

of record pairs required for manual review. In addition to requiring a smaller number of record pairs for clerical review, our model also offers the flexibility of balancing between accuracy and the percentage of record pairs required for clerical review by tuning the threshold value  $T_d$ .

Tables 3 and 4 show that our proposed model achieves higher accuracy (both *AC* and *recall*) than the simple probabilistic model at similar *PP* values for data set 1. For data set 4, it also achieves better or at least comparable accuracy of simple probabilistic model at similar *PP* values. Furthermore, our model has only one parameter  $T_d$ , which is normalised and can be easily set based on the particular application requirement (accuracy and resources available), while the probabilistic model has two parameters  $t_1$  and  $t_2$ , to be set manually. Finding a good set of  $t_1$  and  $t_2$  values without training data is a challenge for many real-world applications. In addition, the probabilistic model requires training data to estimate the conditional probabilities  $m$  and  $u$  for each comparison variable while our model does not need any training data and can directly take the output of any comparison function. Our enhanced model is not restricted to any particular comparison functions and has the potential to be applied directly to the components of the comparison vectors via multi-dimensional clustering. Performance comparison between one-dimensional and multi-dimensional clustering decision models is beyond the scope of this paper and will be the topic of future research.



**Fig. 3.** The *AC* and *PP* values of the probabilistic model for data set 4 under different  $t_1$  and  $t_2$  values.

## 6 Discussion and Conclusions

In this paper, we have reviewed several existing decision models for record linkage and proposed an enhanced clustering-based decision model. Many existing decision models require good training data, which is not readily available in real-world applications. Our enhanced decision model is based on the unsupervised learning technique and does not need any training data. In addition, we have introduced a metric, which can be used to identify record pairs with an uncertain matching status. These record pairs are classified as possibly matched for clerical review. We have also introduced some metrics for comparing the performances of different decision models.

Current experimental results show that the proposed decision model achieves similar accuracy of the existing clustering-based model, but with a much smaller proportion of record pairs for manual review. Furthermore, our model has a mechanism to control the trade-off between accuracy and the amount of clerical review work.

In the current implementation, clustering is performed on the one dimensional feature. Further work is required to test our methodology on the comparison vector components directly using multi-dimensional clustering. We will also look into incorporating clustering algorithms, which can handle categorical comparison values, into our model and test the efficacy of the proposed method to categorical comparison values.

## Acknowledgements

We thank Warren Jin for useful discussions on fuzzy region identification.

## References

1. Fayyad, U., Piatesky-Shapiro, G., Smith, P.: From Data Mining to Knowledge Discovery in Databases (a Survey). *AI Magazine* **17** (1996) 37–54
2. Fellegi, L., Sunter, A.: A Theory for Record Linkage. *Journal of the American Statistical Society* **64** (1969) 1183–1210
3. Winkler, W.: The State of Record Linkage and Current Research Problems. Technical Report RR/1999/04, US Bureau of the Census (1999)
4. Elfeky, M., Verykios, V., Elmagarmid, A.: TAILOR: A Record Linkage Toolbox. In: Proc. of the 18th Int. Conf. on Data Engineering, IEEE (2002)
5. Christen, P., Churches, T.: Febrl: Freely extensible biomedical record linkage Manual. Release 0.2.1 edn. (2003)
6. Elfeky, M., Verykios, V.: On Search Enhancement of the Record Linkage Process. In: Proc. of ACM SIGKDD'03 Workshop on Data Cleaning, Record Linkage, and Object Consolidation, Washington DC, USA (2003) 31–33
7. Gu, L., Baxter, R.: Adaptive Filtering for Efficient Record Linkage. In: Proc. of the SIAM Data Mining Conference. (2004) 477–481
8. Verykios, V., Moustakides, G., Elfeky, M.: A Bayesian decision model for cost optimal record matching. *The VLDB Journal* (2002)
9. Winkler, W.: Using the EM Algorithm for Weight Computation in the Fellegi-Sunter Model of Record Linkage. In: Proc. of the Section on Survey Research Methods. (1988) 667–671
10. Hartigan, J., Wong, M.: A k-means clustering algorithm. *Applied Statistics* **28** (1979) 100–108
11. Fraley, C., Raftery, A.: Model-Based Clustering, Density Estimation and Discriminant Analysis. *Journal of the American Statistical Association* **97** (2002) 611–631
12. Baxter, R., Christen, P., Churches, T.: A Comparison of fast blocking methods for record linkage. In: Proc. of ACM SIGKDD'03 Workshop on Data Cleaning, Record Linkage, and Object Consolidation, Washington DC, USA (2003) 25–27
13. Venables, W., Smith, D.: An introduction to R (<http://www.r-project.org>). (2003)

# Mining Quantitative Association Rules in Protein Sequences

Nitin Gupta, Nitin Mangal, Kamal Tiwari, and Pabitra Mitra

Department of Computer Science and Engineering  
Indian Institute of Technology Kanpur, Kanpur-208016, India  
email: pmitra@iitk.ac.in

**Abstract.** Lot of research has gone into understanding the composition and nature of proteins, still many things are yet to be understood properly. It is now generally believed that amino acid sequences of proteins are not random, and thus the patterns of amino acids that we observe in the protein sequences are non-random. In this study, we are trying to decipher the nature of associations between different amino acids that are present in a protein. This very basic analysis can provide some insight into the co-occurrence of certain amino acids in a protein. Such association rules are desirable for enhancing our understanding of protein composition. They have the potential to give some clue regarding global interactions among particular sets of amino acids occurring in proteins. Presence of strong non-trivial associations further suggests evidence for non-randomness of protein sequences.

## 1 Introduction

Proteins are important constituents of cellular machinery of any organism. Recombinant DNA technologies have provided tools for the rapid determination of DNA sequences and, by inference, the amino acid sequences of proteins from structural genes [1]. The proteins are sequences made up of 20 types of amino acids. Each amino acid is represented by a single letter alphabet, see Table 1. Each protein adopts a unique 3-dimensional structure, which is decided completely by its amino-acid sequence. A slight change in the sequence might completely change the functioning of the protein.

The heavy dependence of protein functioning on its amino acid sequence has been a subject of great anxiety. Research has been done to determine the information content per amino acid in proteins by Yockey [2] and Strait & Dewey [3]. There has been a continuing debate on whether the amino acid sequences of proteins are random or have statistically significant deviations from random sequences. White & Jacobs [5] have shown that any sequence chosen randomly from a large collection of nonhomologous proteins has a 90% or better chance of having a lengthwise distribution of amino acids that is indistinguishable from the random expectation regardless of amino acid type. They claimed that proteins have evolved from random sequences but have developed significant deviations from randomness during the process of evolution. Pande et al [4] mapped protein

sequences to random walks to detect differences in the trajectories of a Brownian particle. They found pronounced deviations from pure randomness which seem to be directed towards the minimization of energy in the 3D structure.

In this study, we take a further step in this direction by trying to predict if there are any co-occurrence patterns among the 20 amino-acids. We have attempted to find out rules that can tell that occurrence of one amino-acid is more likely when another amino-acid is present or absent. Such rules are called “association rules”, and the corresponding technique is called “association rule mining” (ARM). In ARM terminology, the amino-acids may be considered as items, and the protein sequences as “baskets” containing items. See the next section for an introduction to association rule mining. Proteins are polymers of length usually in hundreds. Since the length is much larger, all the 20 amino acids are present in majority of proteins, and thus we will not be able deduce any significant rule just based on presence or absence. To obtain more meaningful association rules in this context, we have incorporated the normalized frequencies of amino-acids observed in each protein, and also discovered “quantitative association rules”, which tell that if one amino-acid A is present with a  $f_1$  frequency, another amino-acid B is likely to be present with  $f_2$  frequency. Our quantitative association rule mining procedure [8] enables us to find these numbers  $f_1$  and  $f_2$ .

The organization of this paper is as follows; the next section gives an overview of association rule mining. Section 3 describes how we have implemented association rule mining for finding quantitative rules in proteins. Section 4 shows the rules that we have obtained. The next section discusses these results and concludes the outcomes of this study. Section 6 on future work describes how this work can be extended. The paper ends with acknowledgment and bibliography.

## 2 Association Rule Mining

Before we begin with the description of our algorithm, it will be helpful to review some of the key concepts of association rule mining. We use the same notation as used in [9]. Let  $I = \{i_1, \dots, i_k\}$  be a set of  $k$  elements, called *items*. Let  $B = \{b_1, \dots, b_n\}$  be a set of  $n$  subsets of  $I$ . We call each  $b_i \subseteq I$  a *basket* of items. For example, in the market basket application, the set  $I$  consists of the items stocked by a retail outlet and each basket is the set of purchases from one register transaction. Similarly, in the “document basket” application, the set  $I$  contains all dictionary words and proper nouns, while each basket is a single document in the corpus. Note that the concept of a basket does not take into account the ordering or frequency of items that might be present. An association rule is intended to capture a certain type of dependence among items represented in the database  $B$ . Specifically, we say that  $i_1 \rightarrow i_2$  if the following two hold

1.  $i_1$  and  $i_2$  occur together in at least  $s\%$  of the  $n$  baskets (the *support*).
2. Of all the baskets containing  $i_1$ , at least  $c\%$  also contain  $i_2$  (the *confidence*).

This definition is also extended to  $I \rightarrow J$ , where  $I$  and  $J$  are disjoint sets of items instead of single items. Let us consider an example of a document basket

S.No.	AA Code	Full-Name
1	A	Alanine
2	C	Cysteine
3	D	Aspartic Acid
4	E	Glutamic Acid
5	F	Phenylalanine
6	G	Glycine
7	H	Histidine
8	I	Isoleucine
9	K	Lysine
10	L	Leucine
11	M	Methionine
12	N	Asparagine
13	P	Proline
14	Q	Glutamine
15	R	Arginine
16	S	Serine
17	T	Threonine
18	V	Valine
19	W	Tryptophan
20	Y	Tyrosine

**Table 1.** Single letter codes of amino acids

application. The baskets in this case are many short stories that are available at our disposal, while the items within each basket are the words. A reader might observe that stories which contain the word “sword” also frequently contain the word “blood”. This information can be represented in the form of a rule as:

$$sword \rightarrow blood$$

$$[support = 5\%, confidence = 55\%] \quad (1)$$

Rule support and confidence are the two measures of rule interestingness [10]. They respectively reflect the usefulness and certainty of discovered rules. A support of 5% for an association rule means that 5% of stories under analysis show that “blood” and “sword” occur together. A confidence of 55% means that 55% of the stories that contain the word “sword” also contain the word “blood”. Typically, associations rules are considered interesting if they satisfy both minimum support threshold and a minimum confidence threshold. Such threshold can be set by users or domain experts. As pointed out in [9], it should be noted that the symbol  $\rightarrow$  is misleading since such a rule does not correspond to real implications; clearly, the confidence measure is merely an estimate of the conditional probability of  $i_2$  given  $i_1$ .

## 2.1 The Apriori algorithm

The most commonly used approach for finding association rules is based on the Apriori algorithm [6]. Apriori employs an iterative approach known as a level-wise search, where  $k$ -itemsets (sets containing  $k$  items) are used to explore  $(k + 1)$ -itemsets. First, the set of frequent (i.e. having more than the minimum support) 1-itemsets is found. This set is used to find set of frequent 2-itemsets, which is used to find the set of frequent 3-itemsets, and so on, until no more frequent  $k$ -itemsets can be found. The efficiency of the level-wise generation of frequent itemsets is improved by using the Apriori property which says that all nonempty subsets of a frequent itemset must also be frequent. This is easy to observe, because if an itemset  $I$  does not satisfy the minimum support threshold, then the set  $I' = I \vee \{i_{new}\}$ , containing all elements of  $I$  and an extra element  $i_{new}$ , cannot occur more frequently than  $I$ , and thus cannot satisfy the minimum support threshold.

## 2.2 Quantitative Association Rules

While the association rule model described above suffices for many applications, it is not adequate when the frequency of each item in the basket is variable and cannot be ignored. For example, in the previously considered example, a user might be interested in the rules of the form:

$$sword_{30-35} \wedge war_{14-16} \rightarrow blood_{50-52} \quad (2)$$

This rule represents that a story that contains between 30 to 35 occurrences of “sword” and 14 to 16 occurrences of “war”, is also likely to contain 50 to 52 references of “blood”. Such rules are called quantitative association rules.

The ARCS system [11] for mining quantitative association rules is based on rule clustering. Essentially this approach maps pairs of quantitative attributes onto a multi-dimensional grid, with the number of dimensions equaling the number of quantitative attributes considered. The grid is then searched for clusters of points, from which the association rules are generated. Techniques for mining quantitative rules based on x-monotone and rectilinear regions were presented in [7]. Approach proposed in [8] works by fine-partitioning the values of the quantitative attributes, and then generating rules of interest.

## 3 Algorithm

Our implementation is based on the partitioning approach described in [8]. We consider 20 attributes in proteins, each related to an amino acid. The value of each attribute in a basket (here protein) is the frequency of the corresponding amino acid in the protein. Since the proteins are of varying lengths, we normalize this frequency by dividing by the length of the protein.

The main steps in the algorithm are as follows:

1. Partition the attributes: We have divided each of the 20 attributes into 10 intervals. In [8], the authors have discussed the notion of partial completeness to quantify the amount of information lost due to partitioning. It has been further shown that for a given number of partitions, equi-depth partitioning (each partition having equal support) gives the minimum loss of information, and is thus optimal. Thus, we have used equi-depth partitioning in our method.
2. The intervals/partitions are mapped into consecutive integers, which are used to represent the intervals. The order of intervals is preserved in the mapping.
3. Find the support for each of the intervals. Also the consecutive intervals are combined as long as their support is less than a predetermined maximum support. This is actually needed in case of equi-distant partitioning when some of the intervals may have very small support and thus it makes sense to combine them with the adjoining intervals. In equi-depth partitioning, all intervals have equal support, and thus this problem does not arise. We identify the set of all intervals which have more than a minimum support *minsup*. This is called the set of *frequent* items. Next we find all sets of items whose support is greater than *minsup*. These are called the frequent itemset, and the algorithm is based on the Apriori algorithm, discussed in the previous section.
4. The frequent itemsets are used to generate association rules. each itemset can give rise to number of association rules by dividing into two parts: antecedents and consequences. For example, an itemset {P,Q,R} can lead to the following rules

- $P \rightarrow Q \wedge R$
- $Q \wedge R \rightarrow P$
- $P \wedge Q \rightarrow R$
- $R \rightarrow P \wedge Q$
- $Q \rightarrow P \wedge R$
- $P \wedge R \rightarrow Q$

The confidence *conf* for each of the rules is determined as the conditional probability of conclusion given precedent. For example, for the rule

$$P \wedge Q \rightarrow R, conf = support\{P, Q, R\} / support\{P, Q\}$$

If the confidence is greater than a pre-determined minimum confidence, *min-conf*, the rule is kept, otherwise it is removed.

## 4 Results

The protein sequences are taken from the SCOP Astral File v1.63 [12], containing only those sequences which are less than 40% homologous to each other. This reduces the bias in favour of highly populated families as compared to sparse ones. The sequences with length less than 100 or more than 500 are not considered.

This gives us a set of 3728 non-homologous amino-acid sequences representing the different types of proteins. In this study our focus is on deriving associations applicable to all proteins in general.

Figure 1 shows the rules obtained with minimum support of 30 proteins. We have obtained 12 association rules, which have confidence more than 50%. For example, the eighth rule indicates that contain very high amounts of Arginine(R) and very low amount of Serine(S) are likely to contain no Cysteine (C). Such rules can provide some insight into the interaction and role of these amino acids in proteins.

Rule	Confidence(%)	Support
$\langle G, 52..500 \rangle \wedge \langle S, 45..500 \rangle \Rightarrow \langle E, 0..16 \rangle$	64.7	33
$\langle E, 0..16 \rangle \wedge \langle L, 0..26 \rangle \Rightarrow \langle T, 40..500 \rangle$	60.9	39
$\langle E, 0..16 \rangle \wedge \langle M, 0..2 \rangle \Rightarrow \langle T, 40..500 \rangle$	59.6	31
$\langle L, 0..26 \rangle \wedge \langle S, 45..500 \rangle \Rightarrow \langle T, 40..500 \rangle$	55.0	38
$\langle E, 0..16 \rangle \wedge \langle L, 0..26 \rangle \Rightarrow \langle G, 52..500 \rangle$	54.6	35
$\langle I, 0..13 \rangle \wedge \langle R, 39..500 \rangle \Rightarrow \langle N, 0..8 \rangle$	54.4	43
$\langle K, 0..11 \rangle \wedge \langle S, 45..500 \rangle \Rightarrow \langle E, 0..16 \rangle$	54.2	32
$\langle R, 39..500 \rangle \wedge \langle S, 0..14 \rangle \Rightarrow \langle C, 0..0 \rangle$	53.5	30
$\langle K, 0..11 \rangle \wedge \langle N, 0..8 \rangle \Rightarrow \langle R, 39..500 \rangle$	53.4	31
$\langle P, 35..500 \rangle \wedge \langle R, 39..500 \rangle \Rightarrow \langle N, 0..8 \rangle$	52.6	30
$\langle L, 64..500 \rangle \wedge \langle P, 35..500 \rangle \Rightarrow \langle N, 0..8 \rangle$	51.7	30
$\langle I, 0..13 \rangle \wedge \langle N, 0..8 \rangle \Rightarrow \langle R, 39..500 \rangle$	50.5	43

**Fig. 1.** Associations obtained using equi-depth partitioning. Each interval (contained in angular brackets) has a amino acid, and frequency range with protein length scaled to 500. The support is the number of proteins in our dataset of 3728 proteins containing all the intervals present in the association rule.

## 5 Discussions and Conclusion

We have used quantitative association rule mining to discuss global associations between amino-acids in proteins. We call the associations global because the rules are not forced to be based on contiguous set of amino acids, and thus can capture global correlations as well. The amino-acid frequencies are divided into intervals to build the rules. We observe that the algorithm gives 12 association rules involving various amino acids.

An important property of our approach is that it can discover rules based not only the presence of amino acids, but also on absence. For example, the eighth rule in Figure 1 has the consequence which says C is likely to be absent. This is a significant difference from the standard motif based works, which are

framed only the basis of presence of an amino-acid. We acknowledge the fact that absence of a particular amino-acid can also be important in the structure and/or function a protein.

To the best of our knowledge, this is the first systematic study to discover global associations between amino acids. This work can be extended in following ways:

- A detailed survey of biological literature regarding proteins can be done to verify the validity of association rules generated by the program. For those rules which are not supported by existing literature, experiments can be conducted to learn more about the interactions in amino-acids.
- Our approach has been based on partitions approach proposed in [8]. It is possible to use other approaches as well, and it is to be seen if they result in some more interesting rules.
- Instead of finding rules based on whole set of proteins, specialized rules can be found for different classes of proteins. This, however, requires a larger protein dataset containing sufficient number of distinct and non-homologous representatives in each class.

## 6 Acknowledgment

We thank Dr. Somenath Biswas (Computer Science, IIT Kanpur) for valuable discussions.

## References

1. Branden, C. and Tooze, J. *Introduction to Protein Structure* (Garland Publishing, New York, 1991).
2. Yockey, H. P. (1977). On the information content of cytochrome. *J. Theor. Biol.* 67, 147-151.
3. Strait, B.J.& Dewey, G.(1996). The Shannon information entropy of protein sequences. *Biophys. J.* 71, 148-155.
4. Pande, S. V., Grosberg, A. Y. & Tanaka, T. (1994). Non-randomness in protein sequences: evidence for a physically driven stage of evolution? *Proc. Natl. Acad. Sci. U.S.A.* 91, 12972-12975.
5. White, S. H. & Jacobs, R. E. (1993). The evolution of proteins from random amino acid sequences. I.evidence of proteins from the lengthwise distribution of amino acids in modern proteins. *J. Mol. Evol.* 36, 79-95.
6. Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules. In *Proc of the 20th Int'l Conference on Very Large Databases*, Santiago, Chile, September'94.
7. Fukuda, T., Morimoto, Y., Morishita, S. and Tokuyama, T. (1996) Data mining using two-dimensional optimized association rules: Scheme, algorithms, and visualization. In *Proc. 1996 ACM-SIGMOD Int. Conf. Management of Data*, pp 13-23, Montreal, Canada.
8. Srikant, R. and Agrawal, R. (1996). Mining quantitative association rules in large relational tables. *Proc. ACM SIGMOD*.

9. Brin, S., Motwani, R., and Silverstein, C. (1997). Beyond market basket: Generalizing association rules to correlations. In *Proc. 1197 ACM SIGMOD*, pp 265-276. Tuscon, AZ.
10. Han, J. and Kamber, M. Data Mining: Concepts and Techniques. *Morgan Kaufmann Publishers*, San Francisco, 2001.
11. Lent, B., Swami, A. and Widom, J. (1997). Clustering association rules. In *Proc. Int'l Conf. Data Engineering (ICDE'97)*, pp220-231, England.
12. <http://scop.mrc-lmb.cam.ac.uk/scop/>

# Mining X-Ray Images of SARS Patients <sup>\*</sup>

XIE Xuanyang, LI Xi, XU Yufeng, WAN Shouhong, and GONG Yuchang

University of Science and Technology of China, Hefei 230027, Anhui, China,  
shiehxy@mail.ustc.edu.cn

**Abstract.** Severe Acute Respiratory Syndrome (SARS), a new infectious disease caused by corona virus, has infected more than 8,000 persons in 32 countries and areas[1] after it first broke out in Guangdong, China, 2002. Another 4 SARS cases were confirmed in China, Jan. 2004. As there were no fast and effective detection method of suspected SARS case, this paper proposes a computer aided SARS detection system (CAD-SARS) based on data mining techniques. During constructing the CAD-SARS system, a training set, as one part of the Picture Archiving and Communication System (PACS) of the 2nd Affiliation Hospital of Guangzhou Medical College, was built to include ‘typical pneumonia’ and SARS X-Ray chest radiographs. Then texture extraction of these images were performed after segmenting out pulmonary fields. Feature vectors were then constructed to build rules for the discrimination of SARS and ‘typical pneumonia’. Two methods were used: decision tree and neural networks, to mine these X-Ray images. Our experiment results showed that more than 70% SARS cases can be detected from normal pneumonia cases. Future works are introduced in this paper.

## 1 Introduction

Severe Acute Respiratory Syndrome (SARS), also called ‘Atypical Pneumonia’ in China, was first found in Guangdong, China, 2002. By July 31, 2003, 5327 patients had been infected with it in China, accounts to 65.6% of all the cases reported in the world[1]. As a newly occurred fast transmittable infectious disease, SARS brings the world not only malady, but the panic caused by knowing little about it. Four newly confirmed SARS cases reported in China this year. It’s VERY important to detect suspected SARS cases early and exactly, the same to the diagnosis of this disease. Aside from this medical importance of fast SARS detection, the detection also has its social meaning for all countries. Confined to the experiences scarcity, especially the subjective experiences, most physicians cannot judge exactly under what condition a patient maybe a SARS suspected case. Although there are some advices in this judging procedure[3], there is a pressing necessity of developing a computer aided detection of SARS system, especially for the countries or regions that have no experience in dealing with this disease.

---

<sup>\*</sup> This work is supported by Science and Technology Bureau of Guangzhou, China

As said in [2][3], besides the epidemiology, diagnostic examine and laboratory set, one of the most important factors of judging a SARS case is the patients X-Ray chest radiographs. Because of the high resolution of X-Ray images ( $2048 \times 2048$  or higher with 12 to 14 bits gray level), confirmation of SARS cases is decided by using X-ray images, especially the Posterior-Anterior (PA) images [4].

Medical Image analysis in combination with data mining yields the possibility of advanced computer-assisted medical diagnosis systems [5]. As many of the research fields are focused in lung cancer [6], breast cancer [7], functional brain image analysis [8], etc., few literatures were found to deal with SARS images. For an inexperienced physician, it's difficult to tell what the differences between these images. Our 'brainstorming' with doctors confirmed this presupposition.

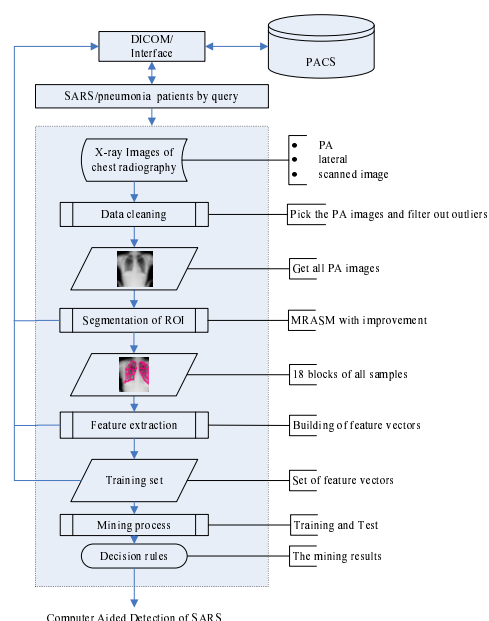
A computerized scheme for early severe acute respiratory syndrome (SARS) lesion detection in digital chest radiographs is presented in [9]. But [9] takes a very strong assumption: early SARS lesion has a spherical shape with linearly decreasing density from its center to border. SARS lesions don't have spherical shape may be ignored. We also note that the SARS detection cannot be fully functioned if no global image information is considered. The data mining techniques can fulfill this target intellectually.

In this paper, a data mining based scheme of Computer Aided Detection of SARS (CADSARS) is proposed. No prior knowledge is needed. The architecture of CADSARS, which is built under the PACS environment, will be given firstly. In Section 3, the data cleaning process is described to construct the high quality training set for the basis of future usage. The definition of Region of Interest (ROI) and how to segment out this area are given in Section 4. The window of lung fields, a important new concept will be introduced in this section too. Section 5 will focused on the feature extraction from ROI and the establishment of feature vectors, which are used as one item in training set. Detailed mining results will be presented in Section 6. Future works and some research in progress are shown in Section 7.

## 2 CADSARS

Availability of Picture Archiving and Communication System raises the possibility of massive digital medical image processing and analyzing. For the detection of SARS from typical pneumonia, an interface between CADSARS and PACS is needed. Because there are variety modalities of images stored in PACS, we need to focus our attention on PA X-ray chest radiography images. This brings the data cleaning procedure. In the literature of pulmonary analysis, each lung field (the left and the right) is divided into 9 blocks. So we need to define the ROI and segment out these blocks. Feature extraction and mining process will be relatively easy if the pre-steps are carried out in high quality. Fig. 2 shows the architecture of CADSARS, which also include concise explanations of all steps.

As mentioned before, a good PACS is the basis of medical image analyzing. The PACS provides an efficient method for storage, transmission and process of medical images. All images are stored as DICOM (Digital Imaging and Com-



**Fig. 1.** Architecture of CADSARS

munication in Medicine) compatible format in PACS. For our mining purpose, only the gray level images are needed. The DICOM/Interface will accomplish the task of exchange data format, the window level/height convert between 12-14 bits bitmap and the ordinary 8-bit gray level image. Because our main task is to distinguish the SARS from typical pneumonia, and there are a variety of modalities (CT, MR, US, SPECT, etc.) stored in PACS, we need a *subset* of all these images: the SARS and pneumonia chest radiographs. This subset is selected randomly and manually.

Three classes of images are included in the *subset*: the Posterior-Anterior images (PA), the Lateral images and the scanned images (examine reports scanned from paper version). Automatic classification method of these three classes is developed. Outliers are filtered out manually because there is no definition of to what degree the 'good' is. Thus the data cleaning process is completed and a high-quality subset of PA X-ray chest radiographs containing typical pneumonia and SARS images is constructed.

Segmentation of ROI is performed in most image analysis systems. In this paper, we use the Multi Resolution Active Shape Model (MRASM) which is an extension of ASM [10]. Also noted by [10], a good starting approximation is important for the convergence of the algorithm. We use the prior knowledge to deduce the initial position and scale parameters, which is proved effective for the algorithm by our experiments. As used by radiologist, the left/right lung field is

divided into 9 blocks separately. The dividing criterion is *outer/middle/inner*  $\times$  *upper/middle/lower*, so totally there are 18 blocks.

Each block's features are extracted based on its statistical and texture calculation. Align all features of a sample image into a row vector to get the feature vector. Each feature vector lies in a very high dimensional space, so the data mining is invoked to build decision rules, the process of reducing the redundancy of data to get knowledge. The mining process is a supervised step since part of the training set is used to train and the other part is used to test the mining result: rules to detect SARS from typical pneumonia.

### 3 Data cleaning

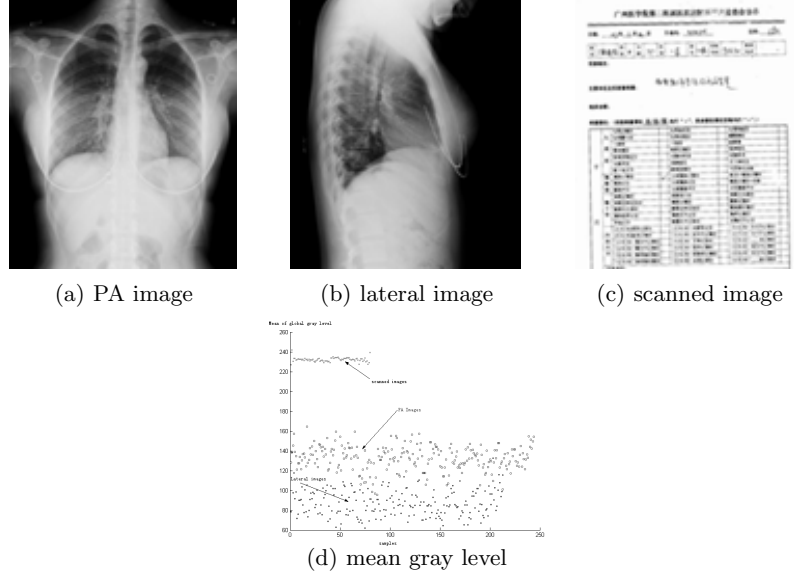
Since real-life data is often incomplete, noisy and inconsistent, pre-processing becomes a necessity[11]. Cleaning is the process of cleaning the data by removing noise, outliers etc. that could mislead the actual mining process. As for image mining on a running PACS system, this pre-process is a must. In our case, the data cleaning process consists of two types of cleaning: the selection of PA X-ray chest images from PACS database and deleting outliers. The last step is carried out manually for some reason explained below.

#### 3.1 Selection of PA X-Ray chest images

Images of all modalities are stored in PACS. For a radiologist, the Digital X-ray images are his focus. The PACS has an hierarchy of privilege control system. The privilege is determined by rank of the doctor belongs to and which department he or she belongs to. Login the PACS as Admin and select all images related to pulmonary disease by the query UI of PACS. Then one physician helps to select the mining oriented images from this relatively small data set. More than 1,000 images were acquired after this semi-automatic selection. Three types of images were left: the PA, the lateral and scanned images of reports. The scanned images of reports are only for the digitalization of all information, so this type of images should be deleted first. A good survey in[6] analyzed lots of papers to show that almost all chest radiography diagnosis were performed in PA images. The overlapping of left and right lung field makes the automatic analyzing very difficult. Although the lateral chest images are need for their specific purpose, they have little usage for our mining goal. All 3 types of images are demonstrated in Fig.2 (a)(b)(c). Fig.2 gives the global mean gray level of these images. For if we denote the image as a matrix  $\mathbf{X}_{mn}$ , where  $m$  equals the height of the image,  $n$  stands for the width, then the global mean gray level of  $\mathbf{X}_{mn}$  is defined as :

$$g_{mean} = \frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n \mathbf{X}_{ij}$$

.The  $g_{mean}$  verses samples is plotted in Fig.2(d). It's obvious that all scanned images are brighter than other types. This is confirmed by Fig.2(a-c). So a straight line can be drawn to delimit scanned images and PA/lateral images.



**Fig. 2.** three types of images and their mean gray level

The PA and lateral images intercross each other in Fig.2(d), a simple linear classification is not enough to separate them. It is observed that if we scale the image  $\mathbf{X}_{mn}$  into a small image, say  $\mathbf{X}'_{50,50}$ , then for a lateral image, either the left column  $\mathbf{X}[1]'$  or the right column  $\mathbf{X}[50]'$  contains only two or three non-zero values. To smooth the sparks, a simple mean filter performs well. This feature can be used to justify whether a image is PA or lateral. Our trial result shows that this simple algorithm cannot reach high precision. Inspired by the deletion of scanned images, we back to use the data mining techniques based on image features. We present the selection of PA images in Algorithm 1.

**Algorithm 1:** Selection of PA images from lateral images

0. select randomly to build a training set for the selection rules building;

1. scale  $\mathbf{X}_{mn}$  to  $\mathbf{X}_{50,50}$ ;

2. calculate the following features:

- $g_{mean} = \frac{1}{50 \times 50} \sum_{i,j=1..50} \mathbf{X}_{ij}$ : global mean gray level
- $meanL = \frac{1}{50} \sum_{i=1}^{50} \mathbf{X}_{i1}$ : mean of the first column
- $meanR = \frac{1}{50} \sum_{i=1}^{50} \mathbf{X}_{i,50}$ : mean of the last column
- $stdL = \sqrt{\frac{1}{50-1} \sum_{i=1}^{50} (\mathbf{X}_{i1} - meanL)^2}$ : standard deviation of the first column

$$- \text{std}R = \sqrt{\frac{1}{50-1} \sum_{i=1}^{50} (\mathbf{X}_{i,50} - \text{mean}R)^2} : \text{standard deviation of the last column}$$

3.set  $\mathbf{v} = (gMean, stdL, meanL, stdR, meanR)^T$  to represent  $\mathbf{X}$ ;

4.let  $N$  denotes for the training sample number, extend  $\mathbf{v}$  with one flag called class:  $Z$  for PA image and  $C$  for lateral image:

$$\mathbf{v}^i = (gMean, stdL, meanL, stdR, meanR, class)^T$$

5.a matrix  $\mathbf{T}$  contains all  $\mathbf{v}^i$  as one row is constructed;

6.using C4.5 as the mining tool to obtain the discriminate rules.

Each  $\mathbf{v}$  is in  $\mathbb{R}^5$ . Decision rules from C4.5 shows that we can project  $\mathbf{v}$  into  $\mathbb{R}^3$ , which is  $\text{Prj}\mathbf{v} = (gMean, stdL, meanR)^T$ . Detailed result will be given in Section 6.

### 3.2 Filter out outliers

Outliers are defined as the images of bad quality, which maybe caused by improper exposure or improper position when photographed. The children images are considered as outliers also because there are images of parents in these images to secure their child stay fixed when taking X-ray images. A few PA images are also deemed as outlier if the images contain pacemaker or other medical attachments. The outliers are filtered out manually to ensure the quality of samples.

## 4 Segment out the ROI

Automatic segment out the Region of Interest (ROI) is virtually mandatory before any computer analysis of X-ray images takes place because the lung field X-ray image contains so much information that we should focus our attention on the mining target: detection of SARS from typical pneumonia. Thresholding, edge detection by masks, region growing algorithms, morphological methods etc are based on the grey level of images. We observed that all these methods are not suitable for the segmentation of ROI in X-ray images, even hybrid methods are inappropriate. Knowledge based segmentation is suitable for the indistinct edge of lung field. Active Shape Model (ASM) [10] and its extension are used widely in medical image segmentation. An accurate initial position placement influences the segmentation result badly. We present our improvement of setting the initial parameters smartly by defining the *window* of ROI, then a simple estimate algorithm is enough to attain high segmentation accuracy.

### 4.1 Definition of ROI

Each PA X-ray of chest radiography comprises two main parts: the left and right lung field as in Fig.2(a). These two parts take up the main part of an image. But

differences exist between images. We define the ROI as these two lung fields. Ideal representation of ROI is to use two close contours. But the analog representation cannot be processed by computer. One simple solution is to use coordinates of key points to express the ROI.

**Definition(ROI)** If there are  $n$  key points, a vector in  $\mathbb{R}^{2n}$ :

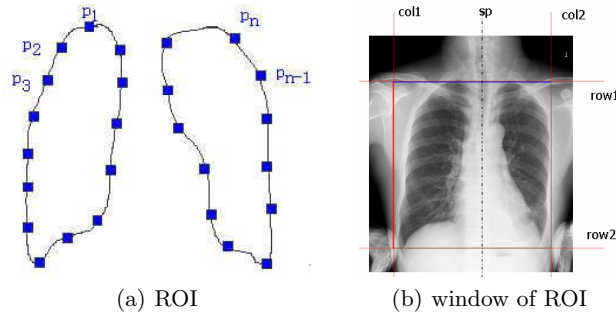
$$\mathbf{x} = (x_1, y_1, x_2, y_2, \dots, x_n, y_n)^T$$

can be used to represent the ROI. Each  $x_i, y_i, 1 \leq i \leq n$  stands for the  $x$  and  $y$  coordinate of key point  $i$ .

**Definition(window of ROI)** The *window* of ROI is ideally defined as the bound-box of  $\mathbf{v}$  which is a 4 parameters vector  $w$ :

$$w = (row_1, row_2, col_1, col_2)^T$$

$row_1 = \min y_i$  and  $row_2 = \max y_i$  denote the upper and lower bound,  $col_1 = \min x_i$  and  $col_2 = \max x_i$  denote the left and right bound, respectively. We will use  $w$  to set the initial parameters of ASM. The definitions of ROI and the *window* of ROI are depicted in Fig.3. Another parameter named SP in Fig.3 defines where the spine column is located in the image. The window of ROI is *ideally* defined because we should get the window *before* ROI is segmented out. It must be obtained in other ways, one of them will be introduced later. In fact, only an approximation to the window can be calculated.



**Fig. 3.** definitions of ROI and window of ROI

#### 4.2 the MRASM algorithm with improvement

The authors of [10] and a series of related papers give the knowledge based algorithm of ASM, a widely used segmentation method. To improve the robustness of ASM, Multi Resolution ASM(MRASM)[12][13] is proposed. In order to use MR search, a pyramid of images with different resolutions should be generated. At the base of the pyramid (Level 0) we have the original image and on higher levels

(Level 1 to L-1) we step-wise decrease the resolution by a factor of two. Not starting the search from the original image (Level 0), MRASM starts at a higher level. See Fig. 4.

MRASM is a supervised model, so a training set is mandatory.

**Definition(training set for MRASM)** Let  $N$  be the sample count, each sample is a ROI  $\mathbf{v}^i$ ,  $1 \leq i \leq N$ , then the training set is defined as:

$$trainingset = \{ \mathbf{v}^i \mid \mathbf{v}^i \text{ is ROI of the } i\text{th sample, } \mathbf{v}^i \in \mathbb{R}^{2n} \}$$

All  $\mathbf{v}^i$  lie in their respective coordinates. In order to compare equivalent points from different ROIs, they must be aligned with respect to a set of axes. Three kinds of shape-invariant operators are enforced upon each  $\mathbf{v}^i$ , all these operators can be written in one formula. If  $\mathbf{v}^i = (x_1, y_1, x_2, y_2, \dots, x_n, y_n)^T \in \mathbb{R}^{2n}$ , then  $\mathbf{v}^i$  is translated by  $t = (dX, dY)$ , rotated by  $\theta$  and scaled by  $s$  of  $\mathbf{v}^i$  can be expressed by  $M(s, \theta)[\mathbf{v}^i] - t$ . The alignment task is to find these parameters:  $t = (dX, dY)$ ,  $\theta$ ,  $s$  that minimize the distance between source  $\mathbf{v}^i$  and target  $\mathbf{v}$ :

$$Err = (\mathbf{v} - M(s, \theta)[\mathbf{v}^i] - t)^T \mathbf{W}(\mathbf{v} - M(s, \theta)[\mathbf{v}^i] - t)$$

where  $bfW$  is weight matrix to give significance to those points which tend to be most 'stable' over the training set. The task of alignment is to align all  $\mathbf{v}^i \in$  training set with respect to a target ROI. Normally the target ROI is the mean shape over training set:

$$\overline{ROI} = \frac{1}{N} \sum_{i=1}^N \mathbf{v}^i$$

Detailed alignment process and results are in [14].

The first and important step in segmentation is how to find initial parameters before any action taken. If we put the initial guess of ROI far away from where the destination ROI is, the hope of ASM converges to ideal target becomes vague, as in Fig. 5.

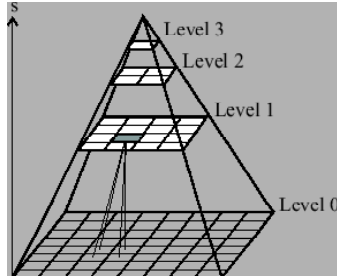


Fig. 4. image pyramid for MRASM

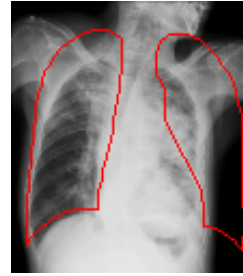


Fig. 5. a divergence example of bad init

There are 4 parameters to be initialized before MRASM algorithm: translation by  $t = (dX, dY)$ , rotation by  $\theta$  and scaling by  $s$ . Because previously we choose

$\overline{ROI}$  as the target of alignment step, set the rotation parameter  $\theta = 0$  is well enough. The translation and scale parameters can be calculated by the following formula provided the *window* of ROI:  $w = (row_1, row_2, col_1, col_2)^T$  has been deduced:

$$(dX, dY)^T = (X, Y)^T - (\overline{X}, \overline{Y})^T$$

$$s = f(s_w, s_h)$$

where:

- $(X, Y)^T$  is the ideal centroid of ideal ROI which can only be approximated by  $(X', Y')$ :

$$X' = \frac{1}{2}(col_1 + col_2) \quad Y' = \frac{1}{2}(row_1 + row_2)$$

- $(\overline{X}, \overline{Y})^T$  is the centroid of  $\overline{ROI}$  defined by:

$$\overline{X} = \frac{1}{n} \sum_{i=1}^n \overline{ROI}_{xi} \quad \overline{Y} = \frac{1}{n} \sum_{i=1}^n \overline{ROI}_{yi}$$

- define the width and height of  $\overline{ROI}$  as:

$$\overline{W} = \max\{x_1, x_2, \dots, x_n\} - \min\{x_1, x_2, \dots, x_n\} \quad \overline{H} = \max\{y_1, y_2, \dots, y_n\} - \min\{y_1, y_2, \dots, y_n\}$$

- define the width and height of *window* of ROI as:

$$W = col_2 - col_1 \quad H = row_2 - row_1$$

- set the scale parameters of X and Y axis as:

$$s_w = W/\overline{W} \quad s_h = H/\overline{H}$$

Because there should be only one scale parameter  $s$  for the reason of shape-invariant, a function  $s = f(s_w, s_h)$  is used to combine  $s_w, s_h$  into one  $s$ . Finally,  $f = 0.9s_h$  is chosen satisfactorily. Only the *window* vector  $w = (row_1, row_2, col_1, col_2)^T$  left to be figured out.

**Left and right bound of window** As mentioned before, only approximation of window can be obtained. If we plot the mean and standard deviation of each column of an image  $\mathbf{X}_{mn}$ , as in Fig.6, the left and right bound of window can be seen easily. The reason is that: when a scan line runs from left to right, first peak is met when scan line enters the right lung field, that's where  $col_1$  is defined; the maximum mean gray level denotes where the spine column is located because of a brighter band in X-ray images; as the column-wise scan line moves to the right,  $col_2$  is met after a decline of the curve. The mm line means 'mean of means' which is equal to  $g_{mean}$ . First and last intersections of mm line with mean curve give values of  $col_1$  and  $col_2$  respectively. In practice, a scale factor of 0.9 is used to enlarge the width of window, because less confinement brings more flexibility.

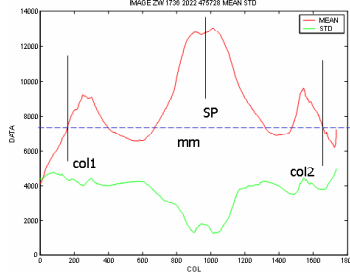


Fig. 6. plot of mean and std of columns

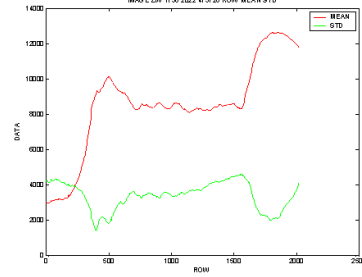


Fig. 7. plot of mean and std of rows

**Upper and lower bound of window** The simple mean of columns plot strategy does not fit the task of finding upper and lower bounds of window, nor the std plot, which can be seen in Fig. 7. No obvious feature can be detected. After a close look, these observations can be made: 1. the X-ray image is approximately symmetric above the shoulders; 2. the lower bound of window  $row_2$  can be calculated easily if  $col_1, col_2, row_1$  are known because for human, ratio of lung fields' height and width is a random number which obeys Gaussian distribution. Chebyshev inequality can be used to assure the accuracy of  $row_2$ . Now the only task is to derive  $row_1$ , which is straightforward if we plot each row's skewness ( $skew = \frac{E(x-\mu_X)^3}{\sigma_X^3}$ ) against row number as in Fig. 8. The vertical dotted line signifies where  $row_1$

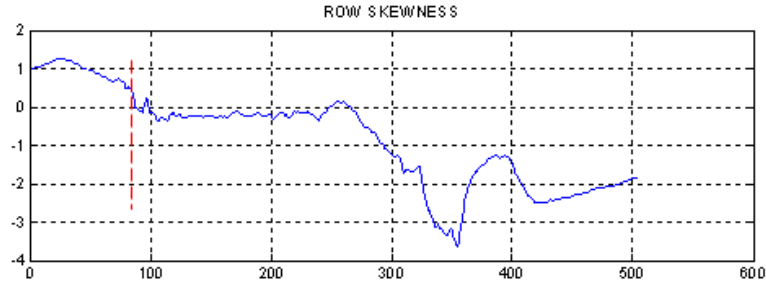


Fig. 8. plot of row skewness against row number

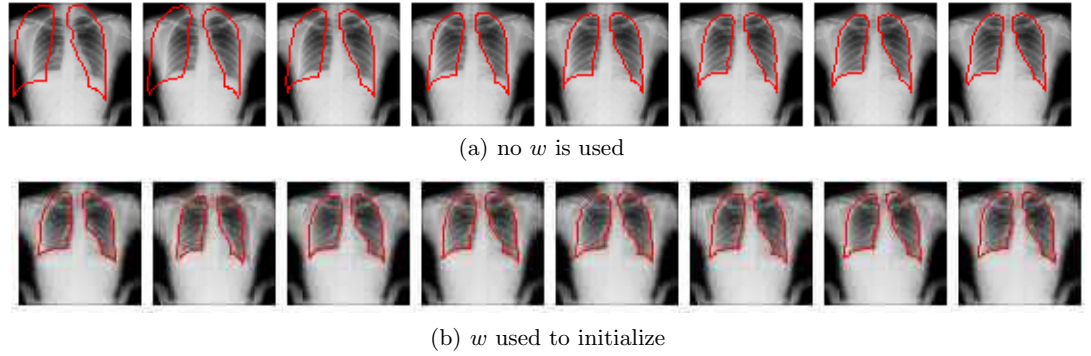
located and also where row skewness turns from positive to negative. Then  $row_2$  is determined by:

$$row_2 = row_1 + (col_2 - col_1) \times (r \pm \delta)$$

where  $r$  represents the ratio of height and width of ROI's window;  $\delta$  is used as an amendatory to achieve accuracy determined by Chebyshev inequality. In our case, experimentation values for  $r$  and  $\delta$  are used.

### 4.3 Segmentation result

Using  $w = (row_1, row_2, col_1, col_2)^T$  to set initial parameters will improve the convergence rate of MRASM. For example, in Fig.9, the first 8 iterations of MRASM are portrayed. Fig.9(a) shows that after about 7 iterations, the algorithm converges to the ideal contour of ROI. But when *window* of ROI:  $w$  is used to initialize translation, scale and rotation parameters, Fig.9(b) reveals that only 1 iteration is needed for ASM's convergence.



**Fig. 9.** comparison of convergence rate between whether *window* is used or not

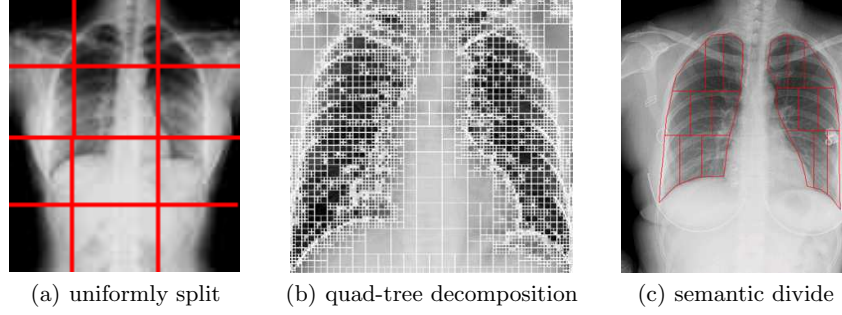
## 5 Feature extraction for mining

Having obtained the ROI, which is a vector  $\mathbf{v}^i \in \mathbb{R}^{2n}$ . Our next task is to divide ROI into individual blocks so that meaningful features for each block can be extracted for mining purpose.

### 5.1 Blocks set of lung fields

Three types of blocks can be used in breaking up lung fields in X-ray images: uniform divide, quad-tree and semantic divide. The simplest way is to split a image equally into four or more blocks and then split further for each block. This uniformly split criterion is used in [15], also shown in Fig.10(a). For chest radiography, this method is so coarse that some important localized information will be lost.

Quad-tree decomposition divides a square image into four equal-sized square blocks, and then tests each block to see if it meets some criterion of homogeneity. If a block meets the criterion, it is not divided any further. If it does not meet the criterion, it is subdivided again into four blocks, and the test criterion is applied to those blocks. This process is repeated iteratively until each block meets



**Fig. 10.** three methods to split a image

the criterion (Fig.10(b)). Because the number of blocks cannot be controlled or estimated by quad-tree decomposition, this method doesn't accord with the mining purpose.

Finally we choose the so called semantic divide to split each lung field into 9 blocks, totally 18 blocks as in Fig.2(c). Splitting criterion is to divide ROI according to Cartesian product:

$$\{upper, middle, lower\} \times \{left, intermediate, right\}$$

.Further split can be found in [16] but our test shows that 18 block is a fairly balance between computation payload and mining results.

## 5.2 Feature vectors

Texture features are extracted from each block. Let  $\mathbf{P}_{m \times n}^{ij}$ ,  $i \in \{0, 1\}$ ,  $1 \leq j \leq 9$  where  $i = 0$  or  $1$  signifies the left or right lung field;  $j = 1, 2, \dots, 9$  signifies each block;  $m \times n$  signifies the size of block  $\mathbf{P}^{ij}$ . For example,  $\mathbf{P}_{50 \times 40}^{1,5}$  defines the 5th block of right lung field, and the block size is  $50 \times 40$ . Of course not all information is useful in the  $50 \times 40$  matrix, only the pixels in ROI is used for feature extraction.

First order and second order texture features are extracted. The first order features include:

1. mean value of gray level:  $MEAN = \frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n \mathbf{P}_{i,j}$ ;
2. standard deviation of gray level:  $STD = \sqrt{\frac{t}{c(c-1)}}$  where:  $t = c \sum_{i=0}^c p_i^2 - \left( \sum_{i=0}^c p_i \right)^2$ , and  $c = mn$ ;
3. skewness of gray level:  $SKEW = \frac{c}{(c-1)(c-2)} \sum_{i=1}^c \left( \frac{x_i - \bar{x}}{s} \right)^3$  where  $\bar{x} = MEAN$  and  $s = STD$ ;

4. kurtosis of gray level:  $KURT = t_1 t_2 - t_3$ , where  $t_1 = \frac{c(c+1)}{(c-1)(c-2)(c-3)}$ ,  $t_2 = \sum_{i=1}^c \left( \frac{x_i - \bar{x}}{s} \right)^4$ ,  $t_3 = \frac{3(c-1)^2}{(c-2)(c-3)}$ ;

Let  $\mathbf{Q}_{k \times k}$  be the concurrence matrix of order  $k$ . Concurrence of four different angles are calculated:  $\mathbf{Q}_{k \times k}^0, \mathbf{Q}_{k \times k}^{45}, \mathbf{Q}_{k \times k}^{90}, \mathbf{Q}_{k \times k}^{135}$ . Then second order features can be extracted :

1. energy:  $ENERGY = \sum_{i=1}^k \sum_{j=1}^k \mathbf{Q}_{i,j}^2$ ;
2. entropy:  $ENTROPY = \sum_{i=1}^k \sum_{j=1}^k (-\mathbf{Q}_{i,j} \log(\mathbf{Q}_{i,j}))$
3. correlation:  $COR = \frac{\sum_{i=1}^k \sum_{j=1}^k ((i-u_x)(j-u_y) \mathbf{Q}_{i,j})}{sig_x sig_y}$ , where  $u_x = \sum_{i=1}^k \left( i \sum_{j=1}^k \mathbf{Q}_{i,j} \right)$ ,  
 $u_y = \sum_{j=1}^k \left( j \sum_{i=1}^k \mathbf{Q}_{i,j} \right)$ ,  $sig_x = \sum_{i=1}^k \left( (i-u_x)^2 \sum_{j=1}^k \mathbf{Q}_{i,j} \right)$ ,  $sig_y = \sum_{j=1}^k \left( (j-u_y)^2 \sum_{i=1}^k \mathbf{Q}_{i,j} \right)$ ;
4. inertia:  $INE = \sum_{i=1}^k \sum_{j=1}^k \left( (i-j)^2 \mathbf{Q}_{i,j} \right)^{\frac{1}{2}}$
5. local calm:  $LC = \sum_{i=1}^k \sum_{j=1}^k \left( \frac{\mathbf{Q}_{i,j}}{1+(i-j)^2} \right)$

For each block  $\mathbf{P}^{ij}$ , 4 first order and 20 second order features (each angle includes 5 features) can be extracted. The for each X-ray image  $\mathbf{X}_{mn}$ ,  $24 \times 18 = 432$  features are extracted. Let  $\mathbf{f}^j \in \mathbb{R}^{28}$ ,  $j = 1, 2, \dots, 18$  denotes feature vector of block  $\mathbf{P}^j$ , then for each X-ray image, define the feature vector as:

**Definition(feature vector):** Define  $\mathbf{F}^i$ ,  $i = 1, 2, \dots, N$  be the feature vector of image  $\mathbf{X}^i$ :

$$\mathbf{F}^i = (ID\#, Imageid, \mathbf{f}^1, \mathbf{f}^2, \dots, \mathbf{f}^{18})^T, i = 1, \dots, N$$

where

$$\mathbf{f}^j = \left( MEAN, STD, SKEW, KURT, \underbrace{\dots}_{0^\circ}, \underbrace{\dots}_{45^\circ}, \underbrace{\dots}_{90^\circ}, \underbrace{\dots}_{135^\circ} \right)^T, j = 1, \dots, 18$$

and  $N$  be the total number in training set.

## 6 Mining for SARS and experiment results

All images in training set can be expressed by its corresponding feature vector  $\mathbf{F}^i$ . After partition all samples for training and testing, outliers which are different from those in Section 3.2 are filtered out automatically. Two mining methods are taken: decision tree and neural networks. Results are given below.

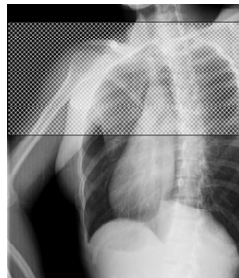
43 SARS patients X-ray images,a total of 241 images and 606 typical pneumonia X-ray images were chosen from PACS for mining.Results of main steps are listed below.

- Automatic selection of PA images from PACS:In 561 images consists of PA,lateral and scanned images,our algorithms Section 3 only falls for 5 cases. A decision tree of 6 leaves is constructed by C4.5.Fig.11 gives each leaves statistical data.Selection rate of 99.16% can be reached.

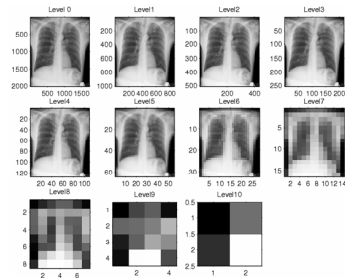
leaf#	samples	Probability of lateral	Probability of PA
4	191	100%	0%
8	14	100%	0%
9	5	40%	60%
10	1	0%	100%
11	6	100%	0%
7	239	0.84%	99.16%

**Fig. 11.** select PA images from PACS

- Compute the *window* of ROI:241 samples were used to test the algorithm proposed in Section 4.2.Result show that only one case(Fig.12) falls.

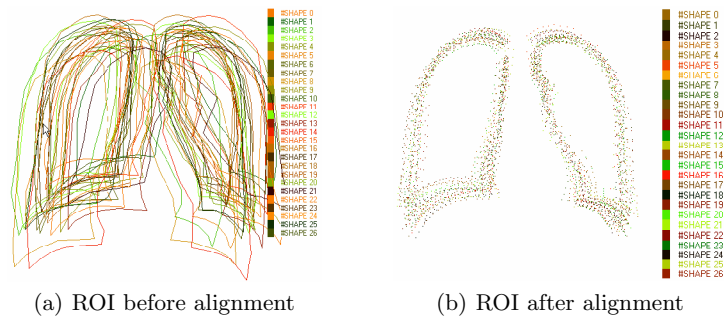


**Fig. 12.** calculation of *window* falls



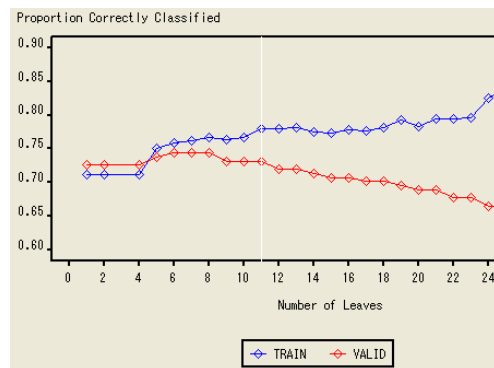
**Fig. 13.** Gaussian pyramid used in MRASM

- Segmentation of ROI:128 key points are used to represent ROI( $n = 128$ ). For ASM,27 training shapes are used.Fig.14(b) is the alignment result of ROIs in Fig.14(a).For each image,a ten level Gaussian pyramid is built for MRASM as in Fig.13.The MRASM with *window*-based initialization segment result is shown in Fig.9(b).

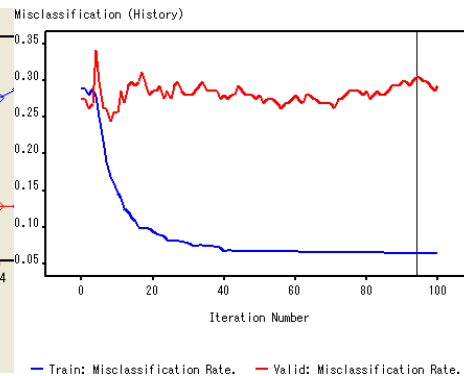


**Fig. 14.** alignment of ROI

- Mining results: C5.0, which upgrade C4.5 slightly, and neural networks were used as two main mining tools. Fig.15 shows that a 11-leaves decision tree gives the best result: rates of correctly detect SARS from typical pneumonia reaches 77.98% in training set and 73.17% when used to justify whether a X-ray chest radiography belongs to SARS or typical pneumonia. At first we think that the neural networks may give better result, but Fig.16 shows that misclassification rate of a NN of three layers with 5 nodes in hidden level is more than 30% when used to classify new images, though the detection rate is 93.6% for training set, which is better than 77.98% of decision tree.



**Fig. 15.** decision tree mining



**Fig. 16.** neural networks mining

## 7 Conclusion and future works

There is an increasing need for the computer aided detection/analysis of SARS. In this paper, CADSARS is proposed to automatic detect SARS from typical pneu-

monia. Directed by CADSARS, main steps of data mining (data cleaning, segmentation of ROI, feature extraction and mining) are discussed. Final results show that the decision tree, which can detect 73.17% of SARS cases from typical pneumonia, performs better than neural networks.

SARS detection plays a central role in our project though the following works are important for further research, some of which are in progress:

1. We have detected SARS images from typical pneumonia images with an acceptable correct rate of 73.17%. More analysis should be made to improve this rate, at the same time to reduce negative rate of misclassification.
2. Pinpoint where lesions of SARS located in a X-ray image of chest radiography. As mentioned in this paper, lung fields are divided into 18 blocks, 9 blocks of each field. The task is to automatically position in which block lesion(s) located. This work is under progress.
3. Only images are used in present research, no diagnosis information of doctors is used. Besides images stored in PACS, examines and diagnosis are also stored in structured report (SR) [17] format. Expert description makes the automatic diagnosis possible if used nicely. For example, text mining technique applies to mining SR data. Combined with feature vectors from images, text feature extraction makes the semi-automatic description generators possible.
4. Image registration and fusion should be used to make the best use of PACS [18]. Only X-ray images (DX) used in present works, but computerized tomography (CT) is also widely used in medical image analysis. Different modalities of images takes different aspects of lesions. Image registration and fusion can bring these information seamlessly together to lay the foundation for data mining and make a promising research area.

## Acknowledgements

We would like to thank those physicians and doctors in the 2nd Affiliation Hospital of Guangzhou Medical College, for providing us the explanation of X-ray images and lists of SARS patients' demography. Also our partners in XHJ. Tech. with whom the PACS are developed rapidly and successfully running for more than 3 years, which makes our research possible. The authors are grateful to all colleagues who help us in programming and providing their comments, advice and support.

## References

1. Summary of SARS case by country, World Health Organization, <http://www.who.int/csr/sars>, 2003.9
2. SARS Reference book (ver.3), Kamps Hoffmann, Flying Publisher, 2003.10
3. Early Advise of Monitoring the Infectious Atypical Pneumonia. the Ministry of Health of China, <http://www.moh.gov.cn>, 2003.11

4. Radiological Appearance of Recent Cases of Atypical Pneumonia in Hong Kong, Anil T. Ahuja et.al. Radiological Department of Chinese University of Hong Kong. <http://www.droid.cuhk.edu.hk>, 2003.10
5. Struggle against cancer inspires PACS data-mining methods. Douglas Page. PACS web <http://www.diagnosticimaging.com/pacsweb>, April 2004
6. Computer-Aided Diagnosis in Chest Radiography: A Survey, Bram van Ginneken et.al, IEEE transactions on Medical Imaging, Vol.22, No.12, December 2001
7. Mammography Classification by an Association Rule-based Classifier. Osmar R. Zaiane et.al, Proceedings of the MDM/KDD 2002, July 2002
8. Data Mining from Functional Brain Image, Mitsuru Kakimoto et.al, Proceedings for MDM/KDD 2000, August 2000
9. A computerized scheme of SARS detection in early stage based on chest image of digital radiograph. Zhong, Z., L. Rihui, et al. Medical Imaging 2004: Image Processing 5370 (Proceedings of SPIE): 904-914
10. Active Shape Models-Their Training and Application. T.F. Cootes, C.J. Taylor, D.H. Cooper, etc.al.. Computer Vision and Image Understanding. Vol.61, No.1, pp.38-59, 1995
11. Data Mining, Concepts and Techniques. Jiawei Han and Micheline Kamber. Morgan Kaufmann, 2001.
12. T. Cootes, C. Taylor, A. Lanitis, Active Shape Models: Evaluation of a Multi-Resolution Method for Improving Image Search. Proceedings of the British Machine Vision Conference, 1994, pp.327-336.
13. Active Shape Models-Part II: Image search and classification. Rafeef Abu-Gharbieh et.al. Proceedings of the Swedish Symposium on Image Analysis, SSAB 1998.
14. Data mining on PACS. Xie Xuan-yang. Technical report of the project (within our team, not published). 2004.8
15. Application of Data Mining Techniques for Medical Image Classification. Maria-Luiza Antonie, Osmar R. Zaiane, Alexandru Coman. Proceedings of the Second International Workshop on Multimedia Data Mining (MDM/KDD'2001)
16. Computer-Aided Diagnosis in Chest Radiography. Bram van Ginneken. Thesis of the author. 2001
17. About SR, Features: The Next Digital Frontier. David J. Vining. 2003-06. Available at <http://www.imagingeconomics.com/library/200306-07.asp>.
18. International Society of Information Fusion, ISIF, <http://www.inforfusion.org/>



## Author Index

N.R.	Achuthan	125	Mehmet A.	Orgun	183
Olatz	Arbelaitz	9	Hung Kook	Park	231
A.	Astigarraga	23	Kang Ryoung	Park	231
Rohan	Baxter	241	Jon	Patrick	33
Lee	Belbin	73	Jesus M	Perez	9
Juno	Chang	231	Tian	Qiu	99
Jie	Chen	157	Gerald	Quirchmayr	149
Peter	Christen	111	Ben	Raymond	73
Tim	Churches	111	Dae Woong	Rhee	231
Raj P.	Gopalan	125	John F.	Roddick	199
Warwick	Graco	183	Marcel va	Rooyen	85
Lifang	Gu	241	Amit	Rudra	125
Nitin	Gupta	255	Wan	Shouhong	263
Ibai	Gurrutxaga	9	B.	Sierra	23
Deepani B.	Guruge	39	Simeon J.	Simoff	55
Markus	Hagenbuchner	215	Byoungcho	Song	231
Yalei	Hao	149	Russel J.	Stonier	39
Hongxing	He	157	Markus	Stumptner	149
Shiying	Huang	169	Kamal	Tiwari	255
Huidong	Jin	157	Ah Chung	Tsoi	215
Chris	Kelman	157	Denise de	Vries	199
Hanseok	Ko	137	Geoffrey I.	Webb	169
E.	Lazkano	23	Graham	Williams	157
Heungkyu	Lee	137	Alan	Willmore	111
Weiqliang	Lin	183	Li	Xi	263
Nitin	Mangal	255	Xie	Xuanyang	263
Jose I.	Martin	9	Hyeon-Joong	Yoo	231
J. M.	Martinez-Otzeta	23	Gong	Yuchang	263
Damien	McAullay	157	Xu	Yufeng	263
Pabitra	Mitra	255	Debbie	Zhang	55
Carl H.	Mooney	199	Shu	Zhang	215
Javier	Muguerza	9	Yihao	Zhang	183
Richi	Nayak	99			

